ACADEMIE MILITAIRE DE TUNISIE

ECOLE ROYALE MILITAIRE DE BELGIQUE

RAPPORT DE PROJET DE FIN D'ETUDES

Thème :

DEVELOPEMENT D'UN ROBOT JOUEUR D'ECHECS

Réalisé par :

L'élève officier

AMIRA Islam

Promotion Habib Thamer

Promoteur **: Professeur BAUDOIN Yvan**

Superviseurs**: M. DE CUBBER Geert**

M. BERRABAH Sid Ahmed

Dédicaces

A la plus précieuse des mamon,

Au plus généreux papa,

A mon grand père ahmed,

A ma chère grand-mère zoubeida,

A mes charmontes sœurs sameh, najla, et amany,

A mon aimable frère hassine et sa femme afraa,

A mes chère beau-frères Lasaad et Mohamed,

A mes mignon neveux saifane et arcelaine,

A toute ma famille de prés et de loin

A tous mes enseignants

A tous mes amis

A tous ceux qui m 'ont aidé à réaliser ce travail

A toutes les personnes que j 'aime et qui m 'aiment

Je dédie ce travail, et qu 'il soit un sentiment de reconnaissance

Honorable et fidèle envers eux.

" Il ne faut jamais avoir peur d'aller trop loin,

car la vérité est au-delà. "

Marcel Proust

Avant propos

Le travail, ici présent, est réalisé au sein du l'Ecole Royale Militaire dans le cadre du projet de fin d'étude pour l'obtention du diplôme d'ingénieur en electromécanique.

Au terme de ce travail, je tiens à remercier mes encadreurs, **Mr. BAUDOIN Yvan** professeur à l'ERM, **Mr. DE CUBBER Geert** et **Mr. BERRABAH Sid Ahmed** chercheurs à l'ERM, qui m'ont honoré tout au long de projet par leurs encadrements de qualité ainsi que leurs soutiens, pour être à la hauteur d'un tel projet.

Aussi, J'exprime toute ma reconnaissance à l'**administration de l'ERM** pour ces efforts majeurs et continus de présenter les meilleures conditions de réussite à ces élèves ingénieurs.

Et toute ma grande gratitude à tous les personnels **de chaire mécanique** et ceux qui n'ont épargné aucun effort à m'aider pour accomplir ce projet.

Un remercie particulier va au **commandant COLON Eric, Mme DOROFTEI Daniela** et **Melle VERBIEST Kristel** pour leurs précieux conseils.

Je voudrais aussi exprimer ma vive reconnaissance envers tous les **enseignants et les personnels de l'Académie Militaire de Tunisie**, ainsi que tout ceux qui ont participés à ma formation.

Finalement, je remercie **les membres du jury** d'avoir accepté d'évaluer ce travail.

ISLAM

Résumé:

changer vers top dow

L'un des objectifs principaux du robot manipulateur industriel SAMSUNG FARAMAN AS2 est de pouvoir prendre des objets (pièces ou outils) et des les déplacer avec plusieurs degrés de liberté dans des locations programmées au préalable vu que son langage de programmation est limité pour quelques objectifs simples. Ce qui fait que on ne peut pas utiliser se genre de robot puisqu'il ne peut pas réfléchir avant d'agir.

L'objet de ce projet est d'introduire de l'intelligence aux déplacements de ce robot affin qu'il puisse s'adapter à jouer d'échec. C'est-à-dire il doit analyser avant l'exécution. Alors, on a eu recours au programme c++ comme programme de contrôle car il est capable de faire la stimulation d'un jeu d'échec.

Pour se faire, nous avons mené on premier lieu une étude théorique et pratique sur robot dans le but de se familiariser et s'adapter avec ses commandes. En second lieu, nous avons utilisé un logiciel d'écoute qui détecte et affiche les commandes envoyées sur le port série avec c++. Par la suite, nous avons copié ces commandes dans un moteur de jeu d'échec, et ainsi nous avons obtenu un robot qui exécute les commandes envoyées par l'ordinateur en analysant l'état du jeu au préalable.

Sommaire

Avant- Propos.....
Table des matières.....
Liste des figures.....
Liste des abréviations .....
Introduction générale .....
CHAPITRE II
CHAPITRE III
CHAPITRE IV
CHAPITRE V

Conclusion et perspectives .....
Annexes .....

Bibliographie .....

Table des figures

Figure 1: sjhdgfsdg 11

Liste des tableaux

Liste des abréviations

CHAPITRE I : INTRODUCTION.

Introduction à la robotique.

La robotique est ensemble de disciplines techniques électromécaniques actionnés par le biais d'un ensemble de logiciels leurs conférant une intelligence artificielle pour servir un objectif communs .cet objectif est l'automatisation flexible de nombreux secteurs de l'activité humaine, l'objet est le robot. D'une part, les progrès réalisés dans les disciplines mentionnées ci haut permettent de construire des robots et autres systèmes mécatroniques performants et fiables. D'autre part, les situations économiques actuelles dans les pays industrialisés, notamment la crise qui a suivi le choc pétrolier des années soixante dis et l'émergence de nouveaux pays industrialisés capable de produire à faible cout ou de fournir rapidement des produit nouveaux et ou de qualité supérieur et assurant la garantie.ces exigences on conduit au développement d'une automatisation flexible impliquant l'implantation de robot dans de nombreuses branches d'activités.

Le nombre des projets de recherche en robotique bénéficiant d'une couverture médiatique importante est croissant.au delà du caractère informatif d'une telle médiatisation, celle-ci entraine des attentes de plus en plus fortes de la part du grand publique et des industriels : les uns se mettent à rêver d'un compagnon domestique, les autres imagines l'usine du future et ses robot dociles. Chacun se construit donc l'image du robot idéal et pense pouvoir retirer un bénéfice de l'utilisation de robots dans son domaine d'activité. De la santé au service passant par la médicine, la défense, la recherche ou la culture, une infinité de possibilités d'utilisation de systèmes robotiques apparait. La réalité des choses est un peut différente. En effet, si la société actuelle est avide de technologie, l'aspect émotionnel d'une

interaction homme robot quotidienne et multiple semble toujours délicat à gérer, surtout pour les hommes. Ainsi, faute de débouché et de démonstration d'une autonomie réelle, la robotique de service reste, à de rares exceptions près, à l'état de démonstrateur technologique et les robots sont aujourd'hui encore utilisés de manière privilégiée pour des missions ou les objectifs sont quantifiables et clairement définis.

De telles missions n'en sont pour autant pas moins complexes et de bras manipulateurs industriels dédiés à des taches précises, simples, dans un environnement bien calibré, nous sommes passés à des robots dont l'espace de travail s'est étendu. Cette extension vient en réponse à la nature des misions qui sont aujourd'hui envisagées en robotique :exploitation, manipulation en milieu hostile(domaine terrestre, sous marin ou spatial),intervention, opération de fabrication sur des pièces de grandes dimensions(soudure, peinture, polissage) ;les exemples ne manquent pas.

1. Définition et historique.

Définition.

Un robot est un dispositif mécanique poly-articulé mis par des actionneurs et commandé par un contrôleur (calculateur) accomplissant automatiquement une grande variété des tâches qui sont généralement considérées comme dangereuses, pénibles, répétitives, impossibles pour les humains ou dans un but d'efficacité supérieure.[]

Le terme robot vient du Tchèque robota, il veut dire le travail forcé ou corvée.

Historique :

- En 1947 : L'apparition de premier manipulateur électrique télé-opéré.
- En 1945 : Invention du premier robot qui peut être programmé.
- En 1961 : Montage d'un robot sur une chaîne de fabrication de General Motors.

- En1961 : Premier robot avec contrôle en effort.
- En 1963 : Le premier contrôle du robot avec vidéo.

2. Les différentes catégories des robots :

- Robots mobiles (figure).
- Robots sous marins (figure).
- Robots volants (figure).
- Robots humanoïdes (figure).
- Robots manipulateurs (figure).

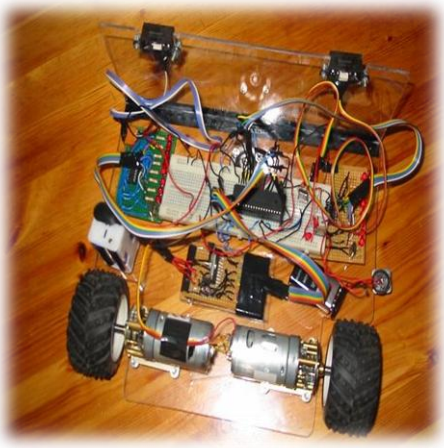
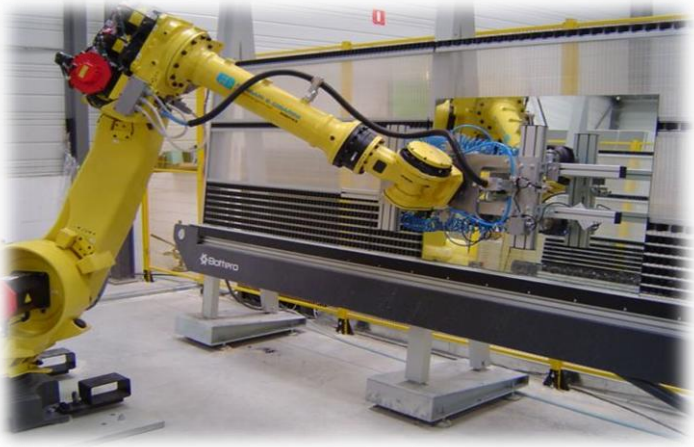


Figure 1: sjhdgfsdg

(figure). Exemple de robot mobil (figure).Exemple de robot sous

Figure.Exemple de robots volants []

(Figure). Exemple de robots humanoïdes



(Figure). Exemple de robots manipulateurs

3. **Vocabulaire utilisé dans le domaine de la robotique.**

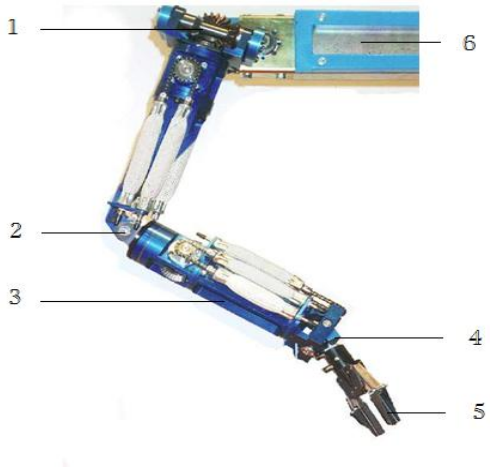


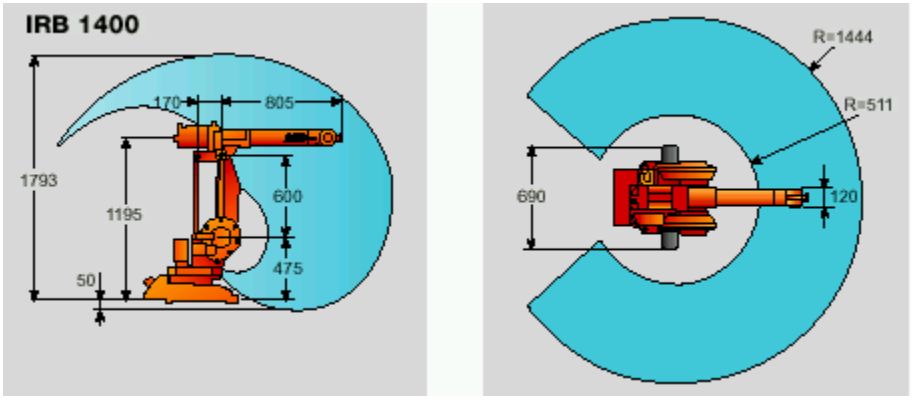
Figure () constituants d'un bras manipulateurs.

1. Actionneur = moteur.
2. Axe = articulation.
3. Corps = segment.
4. Organe terminal.
5. Effecteur = outil.
6. Base.

4. Caractéristiques des robots.

Volume de travail.

Le volume accessible par l'outil du robot est le volume qui peut balayer l'outil. Ce volume dépend de la géométrie du robot, de la longueur des segments et du débattement des articulations (limité par des butées).



(Figure).Exemple du volume de travail d'un robot.

Répétabilité ou précision.

- on dit que le positionnement absolu imprécis lorsqu'il dépasse 1mm, alors on a l'un des erreurs suivantes:

- Erreurs de modèle géométrique.
- Erreurs de quantification de la mesure de position.
- Flexibilité.

- Répétabilité : la répétabilité d'un robot est l'erreur maximale de positionnement répété de l'outil en tout point de son espace de travail.En général, la répétabilité ne doit pas depasser 0.1 mm.

Performances.

- Vitesse maximale :

C'est la vitesse de translation ou de rotation de chaque axe du robot. Souvent les constructeurs donnent une vitesse maximale de l'outil ou l'organe terminal.

- Accélération maximale :

Elle dépend de l'inertie, donc de la position du robot. Pour chaque axe, cette accélération est donnée dans la configuration la plus défavorable (inertie maximale, charge maximale).

Charge utile.

C'est la charge qui garantie une durée de vie la plus longe possible. Donc, c'est la charge maximale qui peut porter le robot sans perturber ni la repètabilité ni les performances dynamiques du robot. Cette charge utile est nettement inférieure à la charge maximale que peut porter le robot et directement dépendante des actionneurs.

5. Différent types de robots.

a. Les robots SCARA.

• SCARA (Sélective Compliance Articulated)

C'est un robot à la fois précis et rapide, utilisé dans l'assemblage

• Caractéristiques :

C'est un robot à 3 axes série (RRT), 3 degrés de liberté, son espace de travail est cylindrique.

Exemple :

(Figure).un robot SCARA.

Les robots cylindriques.

- Ce type de robots est caracterisé par une rapidié et un espace de travail cylindrique.
- 3 axes, série RTT et 3 degrés de liberté.

(Figure).un robot cylindrique.

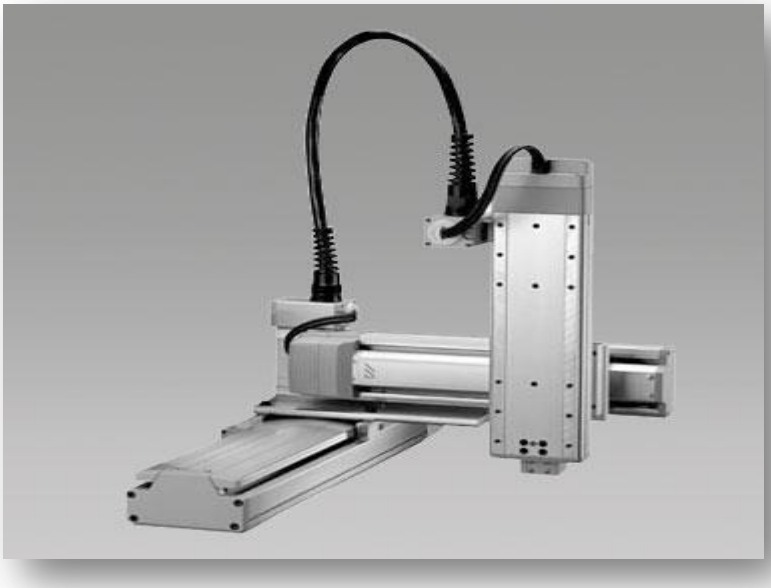
Les Robots sphériques.

- Ce sont des robots à 3 axes, série RRT, et 3 dégrés de libertés, leurs espace de travail est sphérique. Ils sont caractérisés par leur grande charge utile.
- Exemple

(Figure).un robot sphérique.

Les robots cartésiens.

- Se sont des robots à 3 axes perpendiculaires 2 à 2, série, TTT, et 3 degrés de liberté et caractérisés par leur Très bonne précision, mais ils sont lents.
- Exemple :



(Figure). Un robot cartésien.

Les robots parallèles.

- se sont des robots à Plusieurs chaînes cinématiques en parallèles et un espace de travail réduit. Ils sont caractérisés par la rapidité et la précision (grande rigidité de la structure).
- Exemple :

(Figure). Les robots anthropomorphes.

- Leurs structure est produite d'un bras humain. Se sont des robots à 6 axes, série, 6R, et 6 degrés de libertés.
- Exemple :

But du projet

Le but de ce travail c'est de parvenir à contrôler le robot SAMSUNG FARAMAN AS2 pour qu'il puisse déplacer les pièces du jeu dans l'échiquier à partir des solutions choisies par un programme de jeux d'échec exécuté dans l'ordinateur. Et pour atteindre ce but, on est besoin de suivre le plant suivant :

le robot FARAMAN AS2.

1. Introduction du robot.

FARAMAN AS2 est un robot manipulateur série de type anthropomorphe à six axes (six degrés de liberté) et il est composé de chaînes cinématiques poly articulées ouvertes. Ce robot est, en faite, un système de bras mécanique, conçu avec une structure très rependue dans l'industrie.

Avec ces 6 degrés de liberté, l'AS2, exécute les mouvements comme un bras humain .En effet, plusieurs de tâches faites par les humains peuvent être manipulées par le robot AS2.

Le protocole RS232C assure la communication série entre les différentes parties du système constituant le robot. Si on veut commander le robot à distance, la communication ETHERNET est aussi prévue en option.

Pour assurer une exécution fiable, il a été certifié par iso-9001 puisque à chaque étape de sa fabrication, on a respecté les niveaux les plus élevés de la qualité.

LE TP (TEACHING PENDANT) constitue la principale interface avec l'utilisateur car il permet de réaliser plusieurs tâches comme : éditer et exécuter des programmes, gérer les fichiers, positionner manuellement le robot….

En plus de TP un logiciel sous Windows est également disponible sur ordinateur qui permet de réaliser les mêmes fonctions qui est le SRCwin.

- Qui commende le robot ?

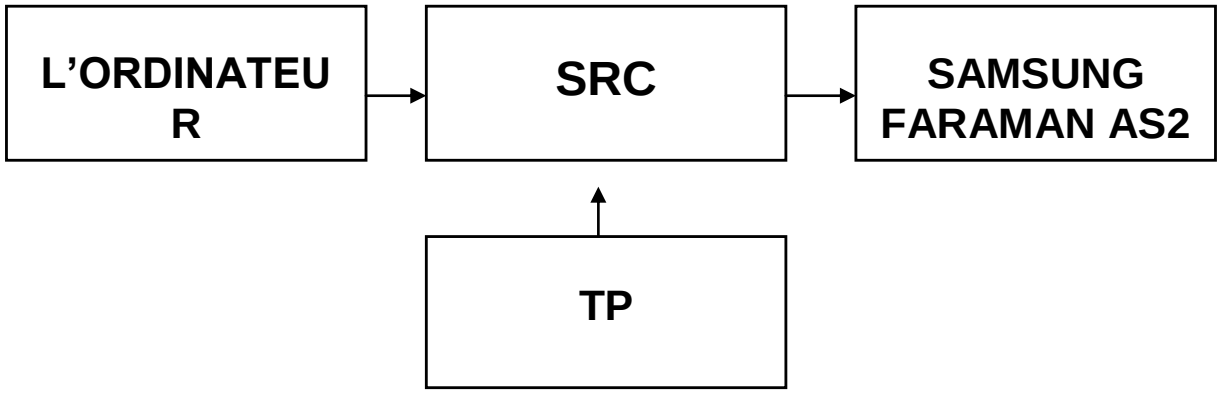


Figure () Schéma du commande du robot.

Comme l'indique la figure, le SRC est l'élément de commende direct du robot par l'intermédiaire soit du TP ou l'ordinateur.

2. **Spécifications de base du robot.**

- type du robot : robot articulé à 6 axes.
- Poids: 58kg.
- Charge utile: 6 Kg.
- Degrés de liberté : 6.
- Précision / Répétabilité: ±1 mm.
- Portée du bras : 858 mm.
- Espace du travail de chaque axe :
  - Axe 1 : ±160°.

- Axe 2 : ±150° ~ -90°.
- Axe3 : ±135° ~ -150°.
- Axe 4 : ±140°.
- Axe 5 : ±120°.
- Axe 6 : ±300°.

- Vitesse max de chaque axe :
  - Axe 1 : 112.5 .
  - Axe 2 : 112.5 .
  - Axe3 : 112.5 .
  - Axe 4 : 225 .
  - Axe 5 : 200 .
  - Axe 6 : 360 .
- Type du moteur : moteur à courant continu équipé d'un cerveau.

3. CONDITIONS D'INSTALLATION :

- Aucun obstacle à linterieur de champs du travail.
- Le contrôleur et le robot ne doivent pas interférer les uns aux autres.
- Le robot doit être visible en cas de mouvement.
- Éviter de mettre le robot dans un endroit humide et poussiéreux.
- La connexion entre un robot et un Contrôleur doit être effectuée après vérification de l'étiquette de câble.
- Vérifier l'état de la connexion du câble interne avant de l'allumer.
- Vérifier l'alimentation électrique du Contrôleur.

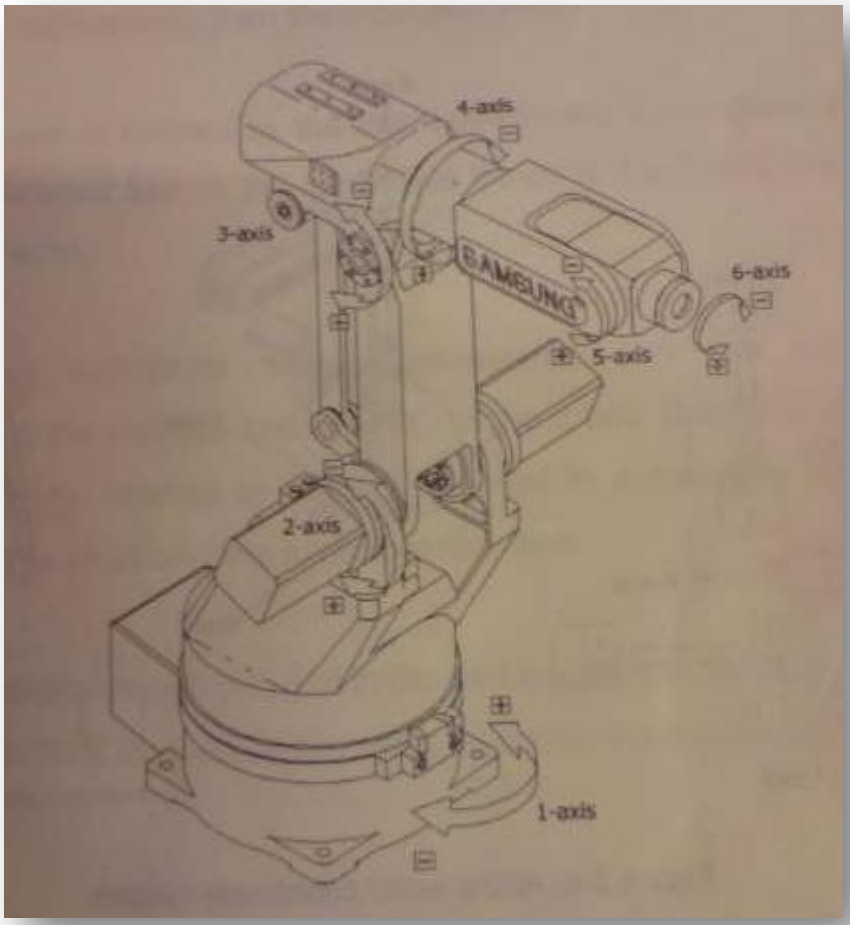


FIGURE ().Les axes du robot SAMSUNG.

4. **Apparences et nomenclature.**

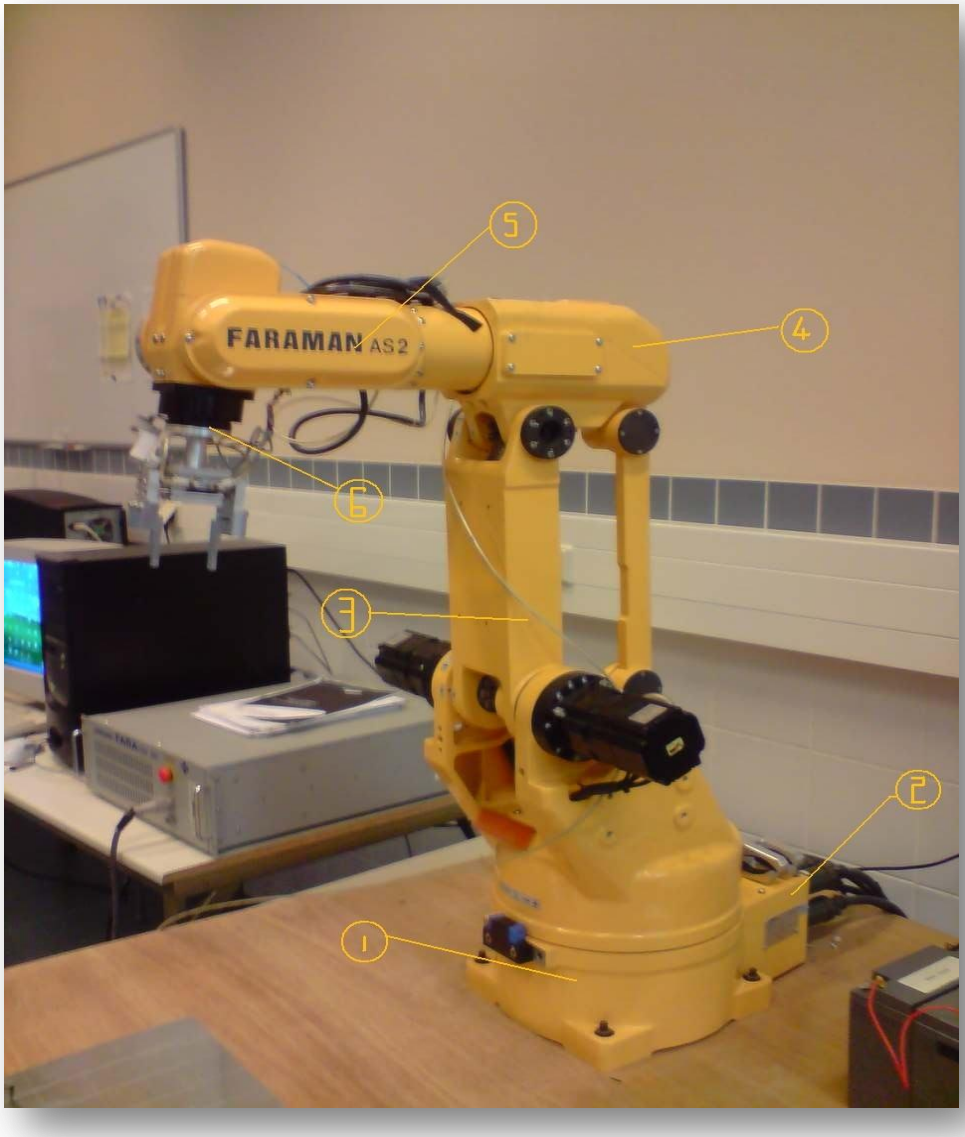


FIGURE () : Constituants d'un bras manipulateur.

- Base.
- Coffret d'extrémité.
- bras inférieur.
- bras supérieur.
- Poignet.
- Terminal.

5. **Dimensions et espace du travail.**

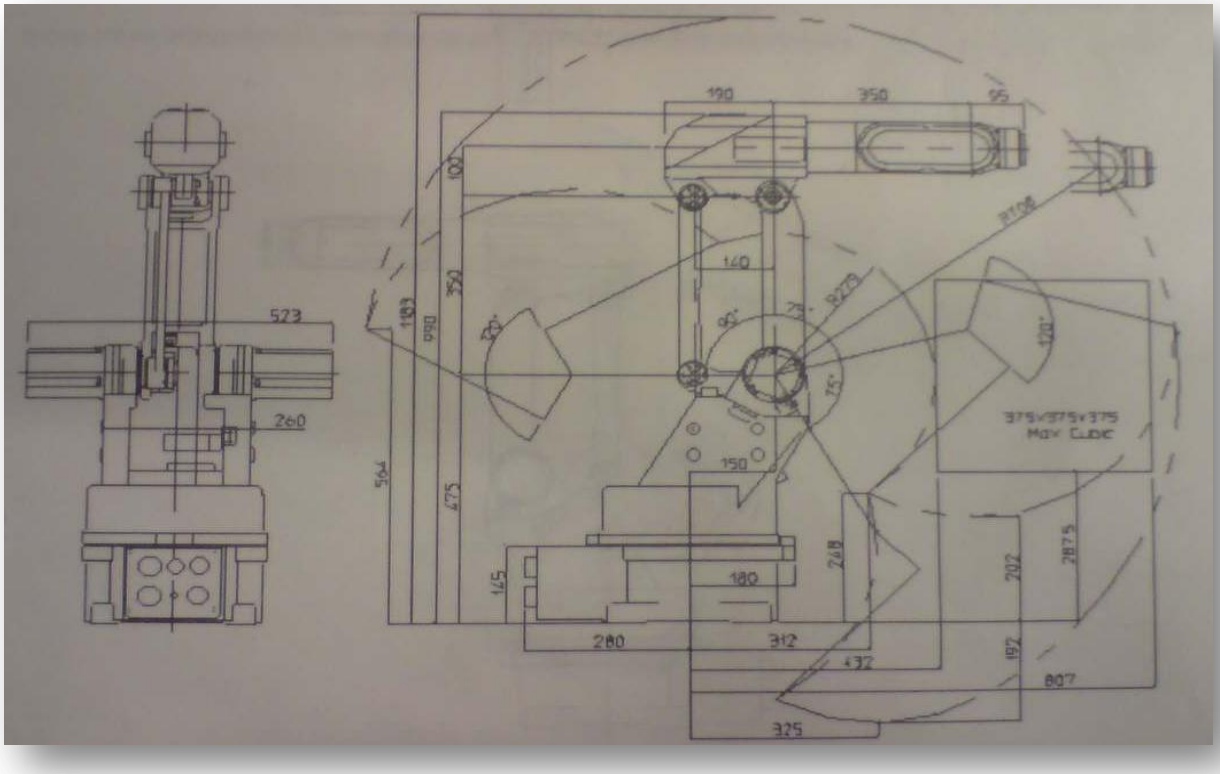


FIGURE () plan de travail vertical.

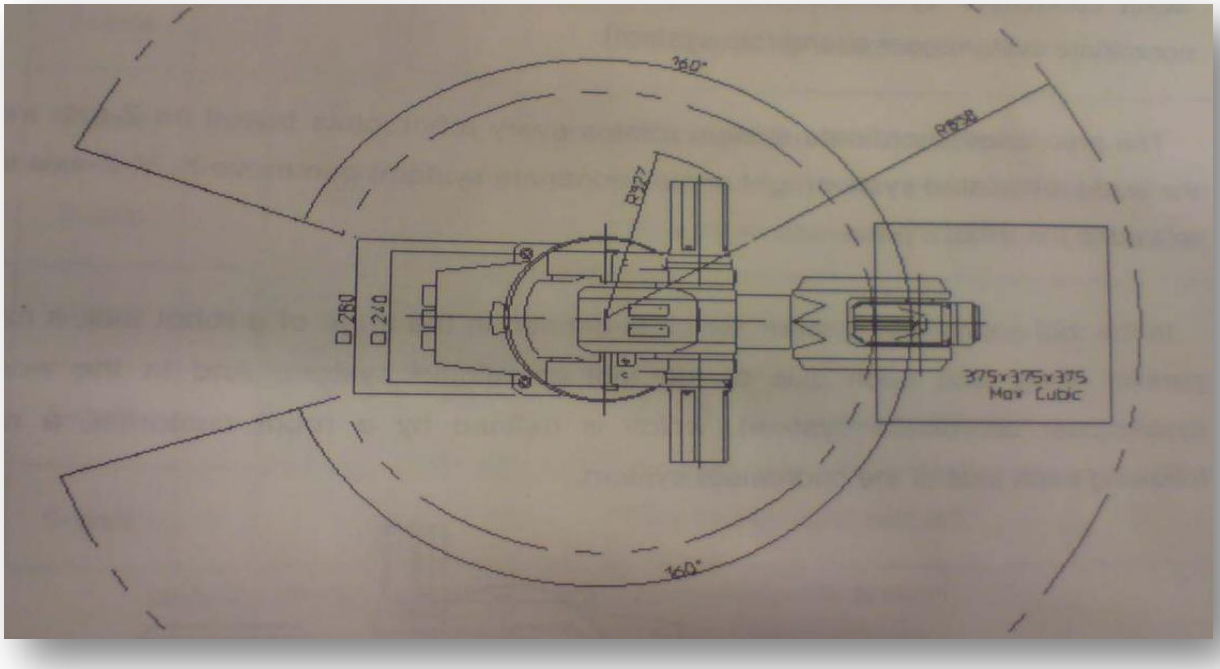


FIGURE () plan de travail horizontal.

6. **Les différents systèmes d'axes.**

Suivant les besoins de l'utilisateur et la situation auquel il est mis, les mouvements du robot peuvent être décrits dans quatre systèmes de coordonnes, et qui sont les suivants :

- Système des coordonnées cartésien (world coordinate system) :

Les mouvements du robot peuvent se faire parallèlement à chaque axe.

- **Système des coordonnées outil** (TOOL COORDINATE SYSTEME) :

Le déplacement du robot ce fait parallèlement aux axes du poignet.

- **Système des coordonnées articulé** (articulated coordinate system) :

Chaque axe peut tourner indépendamment des autres axes, la rotation de chaque articulation se fait indépendamment des autres.

- Système de coordonnées utilisateur (work coordinate system) :

Le robot se met en mouvement dans un système de coordonnées définies par l'utilisateur.

7. **Moment et d'inertie admissible au poignet.**

Axe 4	**moment de roulement admissible M_4(N.M)**	**13**
Axe 4	inertie de roulement admissible J_4 (Kg .M^2)	0.23
Axe 5	moment de flexion admissible M_5(N.M)	9.4
Axe 5	l'inertie de flexion admissible J_5 (Kg .M^2)	O.18
Axe 6	moment de torsion	5.9

	admissible M_6(N.M)	
Axe 6	l'inertie de torsion admissible J_6 (Kg .M^2)	0.06

Tableau : Moment et d'inertie admissible au poignet.

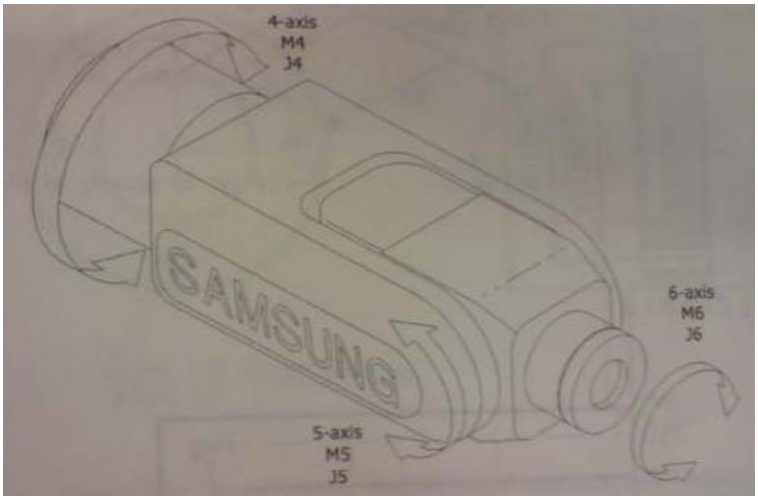


FIGURE : Les axes dans le poignet.

Suite du texte

Dans le premier chapitre, nous rappellerons quelques notions liées à la robotique (vocabulaire de la robotique, catégories des robots, caractéristiques et types des robots). Ainsi, après avoir énoncé ces définitions, on va introduire et définir notre robot à partir de ce qui précède.

Dans le deuxième chapitre, nous énoncerons analyse cinématique et modélisation mathématique d'un robot .et a la fin on va essayer d'appliquer cette étude a notre robot

Dans le troisième chapitre nous allons résumer comment commander le robot par les moyens qui existes (TP, ordinateur).puis nous allons essayer de le commander par c++ en utilisant les commandes précédentes, un logiciel d'écoute sur port série.

Dans le quatrième chapitre, nous parlons des moteurs de jeu d'échec et les algorithmes choisies pour et on finis par choisir le moteur qui convient a notre cas.

Dans le cinquième chapitre, nous utiliserons tout ce que précède dont le but d'arriver à coordonner entre le moteur du jeu d'échec et les mouvements du robot. Et on termine par éprouver notre réussite.

Chapitre 2 : théorie de contrôle du robot

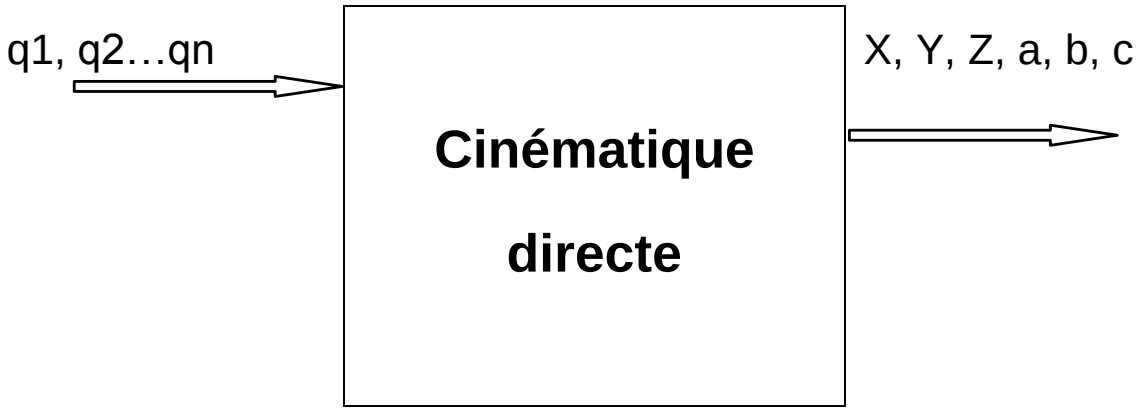
Pour commander notre robot par le SRC, il est nécessaire de posséder son modèle.si on connait la configuration du robot, le modèle géométrique directe (MGD) nous permet de déterminer se situation alors que le modèle géométrique inverse (MGI) permet de déterminer sa configuration à partir de sa situation.

Le but de ce chapitre est de déterminer le modèle géométrique directe puisque le robot nous donne le modèle géométrique inverse en affichant à chaque fois l'ensemble des angles articulaires a partir d'une position et une orientation de l'outil par rapport à la situation de travail.

I. Analyse cinématique et modélisation mathématique d'un robot

1. Problème cinématique direct :

La cinématique directe consiste à trouver la position du centre de l'outil et son orientation en fonction des paramètres donnes a l'entrée (coordonnes articulaires).



Ou

- q1, q2…qn : sont les coordonnées articulaires.

- n : le nombre de degrés de liberté.

- X, Y, Z : les coordonnés du TCP.

- a, b, c : sont les angles donnant l'orientation de l'outil.

Pour une machine a trois degrés de liberté voir figure (robot sphérique) en exprimant X, Y, Z en coordonnées sphérique, le problème se résout facilement.

Voir figure (robot cartésien).

Pour un autre robot a trois degrés de liberté cartésien robot on voit que q1, q2, q3 égale respectivement à X, Y, Z.
Lorsque le degré de liberté dans un robot augmente, le problème devin plus difficile à résoudre. Pour résoudre se problème on a choisie la méthode de **Denavit-Hartenberg** qui est la plus utilisée et cette méthode sera détaillée dans ce qui suit.

2. **modélisations**

On considère un bras articulé, on peut la décomposer en n(=degrés de liberté) segments, le segment zéro correspond a la base.

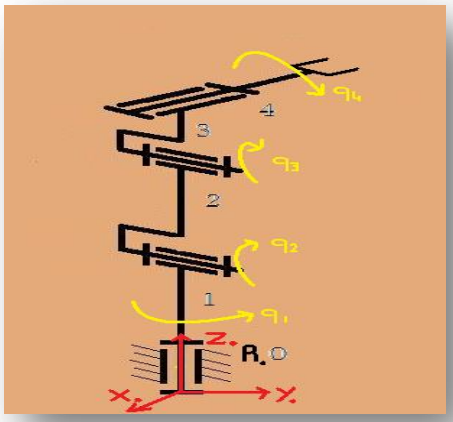


Figure : Exemple d'un bras articulé.

Si au segment i, on fixe un repère Ri =(o_i ,x_i , yi , z_i) et au segment i+1, un repère R_{i+1}=(o_{i+1} , x_{i+1}, y_{i+1} , z_{i+1}),on constate que le repère i+1 subit dans le repère i soit une rotation soit une translation .or, ces deux mouvement peuvent être aisément exprimes a l'aide de matrice homogène de transformation :$T_{i,i+1}$.
Ces matrices sont telles que : si A_{i+1}= (x_{i+1}, y_{i+1} , z_{i+1}) représentes un vecteur dans le repère i+1,alors A_i = $T_{i,i+1}$*A_{i+1}represente le même point mais exprimes dans le repère i. Ai est donc l'image de Ai+1par la transformation $T_{i,i+1}$.

a. **combinaison du mouvement de rotation et de translation**

Si le repère fait au même temps une rotation d'un angle α et une translation du vecteur T par rapport au repère 1 comme le montre la figure

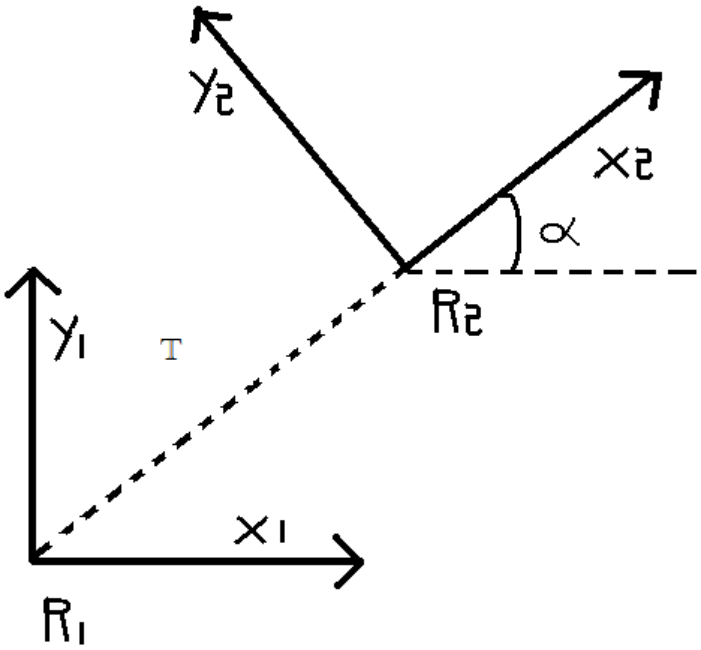


Figure : mouvement de rotation et de translation du repère 2 par rapport au repère 1
On aura la matrice de transformation de la forme :

T_{12}=

cos α	**-sin α**	**T1**
sin α	**cos α**	**T2**
0	**0**	**1**

Si on travail dans un espace a trois dimensions, il en va de même et on obtient un matrice de transformation entre le repère i+1et le repère i :

$T_{i, i+1}$=

			T1
	ROTATION		**T2**
			T3
0	**0**	**0**	**1**

Ou la matrice la matrice rotation (3x3) exprime la rotation qu'a subi le repère i+1par rapport au repère i.

-Signification de la matrice :

Les termes de la première colonne de rotation sont les coordonnées du vecteur unitaire x_{i+1}, exprimées dans le repère i ; ceux de la deuxième colonne celles de y_{i+1} et ceux de la troisième celles de z_{i+1} .Quand à T1, T2, T3, ce sont les coordonnes exprimées dans le repère i du vecteur joignant les deux origines Oi et Oi+1.

Il est donc possible d'exprimer les coordonnes d'un point du repère i+1 dans le repère i si l'on connait $T_{i,i+1}$.

Ainsi, dans le cas d'un bras articulé à six degrés de liberté, les coordonnées du point du centre de l'outil (TCP) dans le repère absolu sont :

$TCP_{absolue}=T_{01}*T_{12}*T_{23}*T_{34}*T_{45}*T_{56}$

On a donc une méthode modélisation mathématique d'un robot. Pour prévoir la mettre en œuvre, il faut prévoir définir de façon logique les repères lies à chaque segment. Pour ce faire, on applique la technique de **Denavit-Hartenberg**

3. **Paramètres de Denavit-Hartenberg modifiés**

a. **Méthode**

Il est possible de modéliser un bras articulé comme l'on a constaté dans le paragraphe précédant en exprimant que chaque repère lié à un segment a subit une rotation et une translation par rapport au repère lié au segment précédent.

La méthode Denavit-Hartenberg modifiés est la technique la plus simple à utilises et la plus rependue pour nous informer sur la géométrie d'un bras manipulateur, et pour la métriser il sufi seulement de bien comprendre les *paramètres* de Denavit-Hartenberg modifiés, cette méthode propose des façons simples et efficace, les matrices de transformation $T_{i,\ i+1}$ entre deux segments successifs ainsi que les repères qui y sont liés.

Cette méthode est applicable pour tout bras articulé ouverte, ou chaine cinématique ouverte, dont chaque articulation possède un degré de liberté de rotation ou de translation. Ceci couvre la majorité des robots même notre robot.

Procédure :

Numérotation des segments

on commence a numéroter les segments constitutifs du bras manipulateur en commençant par la base, jusqu'a arriver à l'outil qui va porter l'indice n (n=6 dans notre robot) égale au nombre de degrés de libertés .

On définie le système d'axes de chacun des segments

Pour définir le système d'axes des segments en suit les règles suivantes :

On définie le repère absolue $R_0 = (O_0, x_0, y_0, z_0)$ de manière arbitraire (fixe), le repereR_0 est défini librement en suivant des considérations de simplicité, il faut simplifier la matrice de passage entre l'élément 0 et 1.

De même, le point On est choisi de façon à annuler un maximum de paramètres. Un point O_{n+1} est associé à l'organe terminal du bras manipulateur, selon des critères opérationnels :

Typiquement, si le bras manipulateur est équipé d'une pince, le point O_{n+1} représentera a la fois celle a laquelle on attribue la plus grande généralité et la plus intuitive en pratique.

Pour placer les repères R_1 à R_{n-1} selon la règle des paramètres de Denavit-Hartenberg modifiés, il convient de suivre les réglés suivantes, pour i = 1,2, 3, .., n:

_ placer les axes Z_i (i=1….n) dans l'axe de rotation de chaque charnière ou dans l'axe de translation de chaque glissière.

– O_{i-1} est le pied de la perpendiculaire commune aux axes des liaisons L_{i-1} et Li, situé sur $L_{i-1.}$

Si les axes des liaisons sont parallèles, il faut choisir arbitrairement une perpendiculaire commune.

– choisir chaque axe Xi perpendiculairement à Zi et Zi+1. Si ceux-ci se coupent, l'origine Oi sera ce point d'intersection et Xi=Zi x Zi+1, sinon Xi sera placé sur la perpendiculaire commune à zi et Zi+1 et orienté suivant la règle du tire-bouchant. Si Zi et Zi+1sont confondues, Xi se place n4importe ou dans un plan perpendiculaire à zi et Zi+1.

Si les axes des deux liaisons sont concourants ou confondus, l'orientation est arbitraire (règle avant, droite, haut) ;

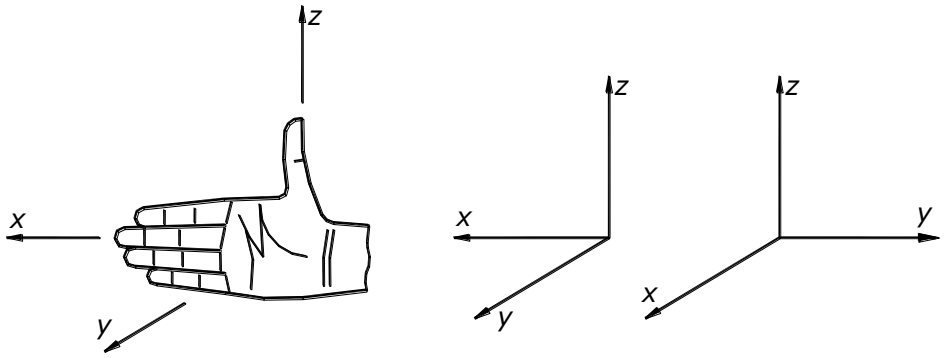


Figure : comment choisir l'axe des y.

– y_{i-1} est tel que le repère R_{i-1} soit orthonormé direct comme le montre la figure précédente.

Les Paramètres de Denavit-Hartenberg modifies :

Compte tenue de tout qui précède textuellement, on peut les définir comme suit :

– α_{i-1} : appelé « torsion » angle algébrique entre z_{i-1} et z_i, mesure autour de x_{i-1} .

– a_{i-1} : appelé « longueur », est la distance arithmétique de la perpendiculaire commune aux axes des liaisons L_{i-1} et Li mesurée le long de x_{i-1} ;

– θ_i : angle algébrique entre x_{i-1} et xi, mesure autour de z_i .

– d_i : appelé « débattement », est la distance algébrique du point O_i à la perpendiculaire, mesurée le long de zi.

Les paramètres de Denavit-Hartenberg modifies permettent de déterminer de manière unique la configuration des corps et donc la géométrie du bras manipulateur. Pour visualiser plus facilement les paramètres et annuler certains paramètres constants il est souhaitable lorsqu'on modélise un bras manipulateur de présenter celui-ci dans une configuration particulière.

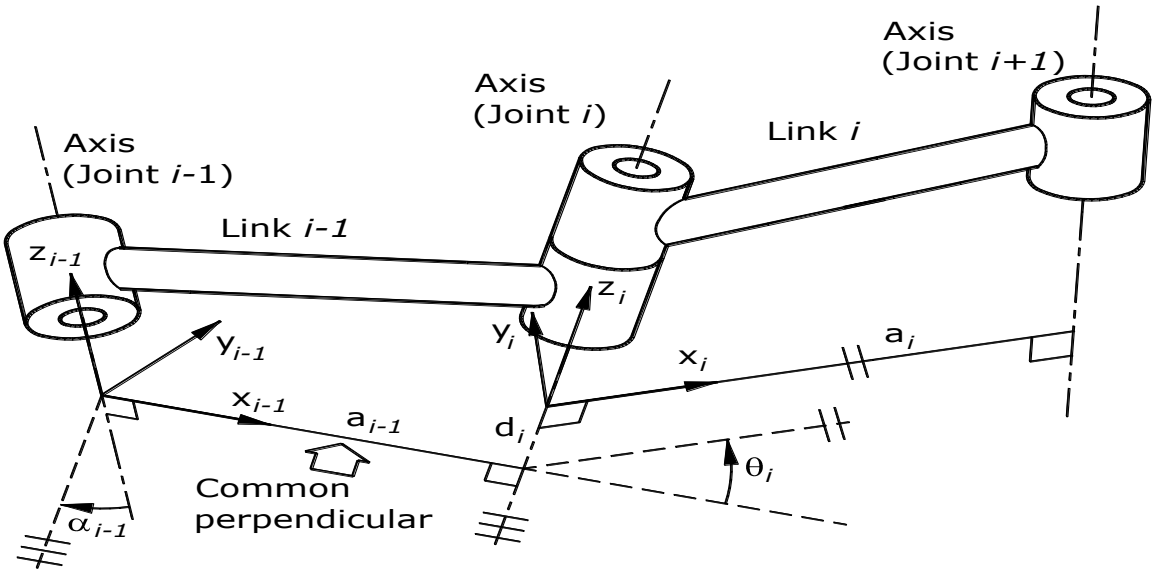


Figure : Explication des paramètres de Denavit-Hartenberg.

Application de Denavit-Hartenberg au robot SAMSUNG FARAMAN AS2

1. PREMIERE ETAPE.

Le schéma cinématique simplifié du robot est le suivant :

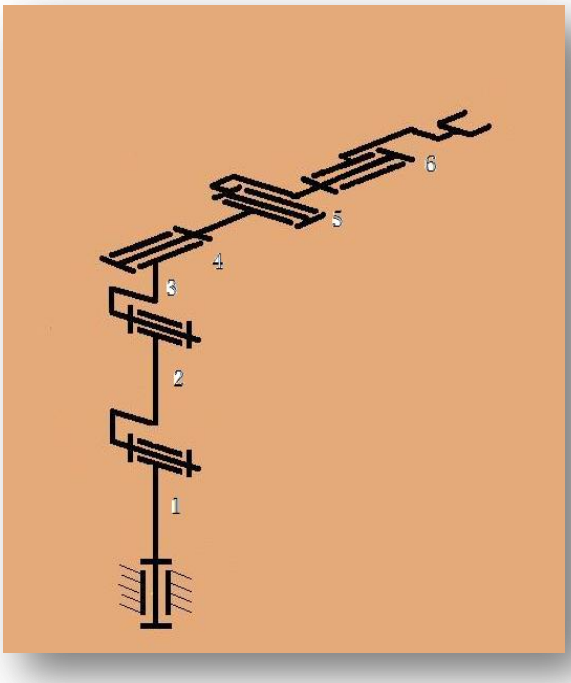


Figure : numérotation des segments

2. **Deuxième étape.**

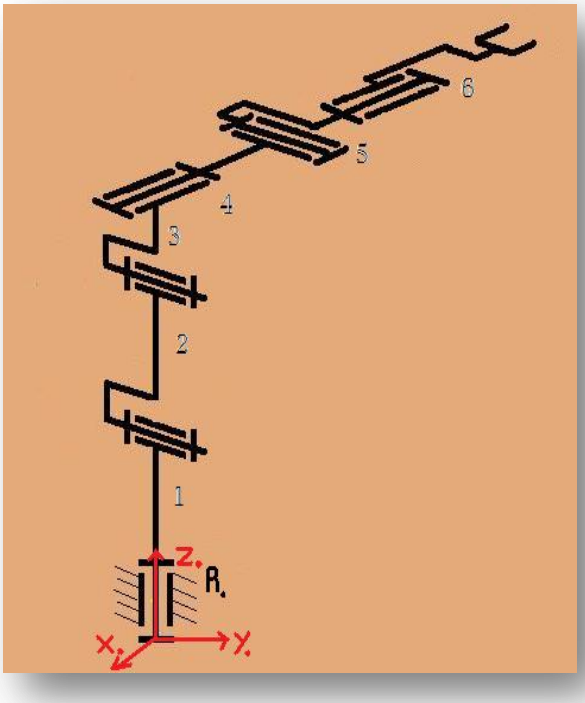


Figure : Définition du repère r_0 d'une manière arbitraire

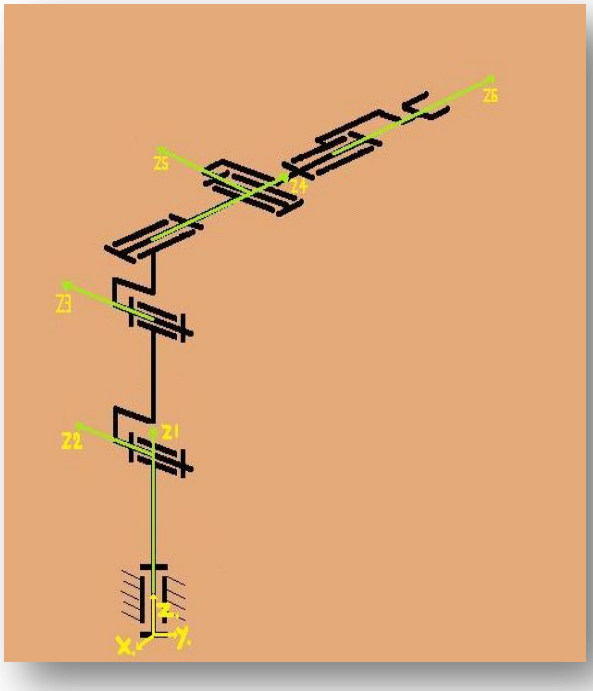


Figure : Choisir les z_i

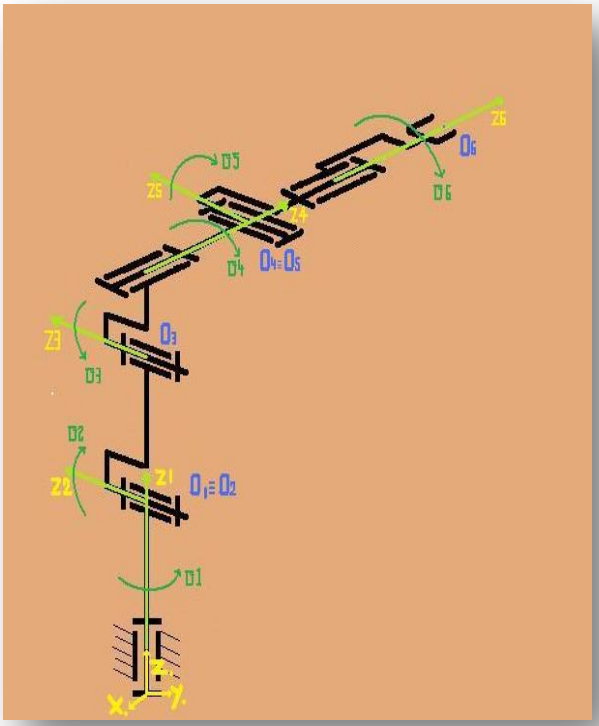


Figure : Choisir les origines o_i :

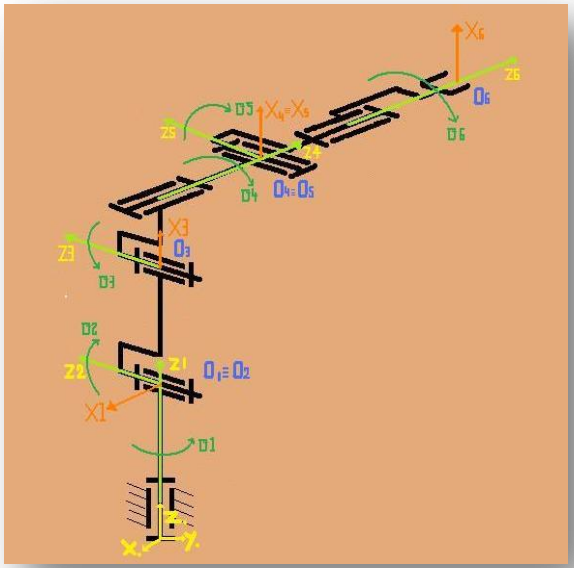


Figure : Choisir les x_i :

TROISIEME ETAPE

i. *Tableau de Denavit-Hartenberg*

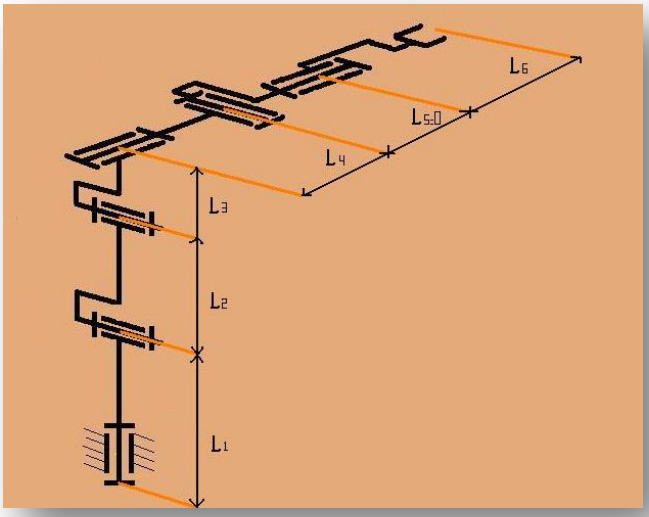


Figure : les segments du robot.

ELEMENT MOBILE	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	L1	θ 1
2	0	π/2	0	π/2+θ 2
3	L2	0	0	θ 3
4	L3	- π/2	L4	θ 4
5	0	π/2	L5=0	θ 5
6	0	- π/2	L5+L6=L6	θ 6

Tableau : Le tableau de Denavit-Hartenberg du robot.

Avec :

L1=330, L2=350, L3=100, L4=350, L5=O, L6=112.5

LES MATRICES DE DENAVIT HARTENBERG :

$T_{01}=$

T_{12}

$T_{23}=$

T_{34}=

T_{45}=

T_{56}=

T06= T01* T12* T23* T34*T45*T56

En effectuant la multiplication de ces matrices par MATLAB on obtient une matrice de la forme :

T06=

Avec :

A =

(((cos(O1)*cos(O2+90)*cos(O3)-cos(O1)*sin(O2+90)*sin(O3))*cos(O4)-
sin(O1)*sin(O4))*cos(O5)+(-cos(O1)*cos(O2+90)*sin(O3)-
cos(O1)*sin(O2+90)*cos(O2))*sin(O5))*cos(O6)-((cos(O1)*cos(O2+90)*cos(O3)-
cos(O1)*sin(O2+90)*sin(O3))*sin(O4)+sin(O1)*cos(O4))*sin(O6)

B=

-(((cos(O1)*cos(O2+90)*cos(O3)-cos(O1)*sin(O2+90)*sin(O3))*cos(O4)-
sin(O1)*sin(O4))*cos(O5)+(-cos(O1)*cos(O2+90)*sin(O3)-
cos(O1)*sin(O2+90)*cos(O2))*sin(O5))*sin(O6)-((cos(O1)*cos(O2+90)*cos(O3)-
cos(O1)*sin(O2+90)*sin(O3))*sin(O4)+sin(O1)*cos(O4))*cos(O6)

C=

-((cos(O1)*cos(O2+90)*cos(O3)-cos(O1)*sin(O2+90)*sin(O3))*cos(O4)-
sin(O1)*sin(O4))*sin(O5)+(-cos(O1)*cos(O2+90)*sin(O3)-
cos(O1)*sin(O2+90)*cos(O2))*cos(O5)

X=

(-((cos(O1)*cos(O2+90)*cos(O3)-cos(O1)*sin(O2+90)*sin(O3))*cos(O4)-
sin(O1)*sin(O4))*sin(O5)+(-cos(O1)*cos(O2+90)*sin(O3)-
cos(O1)*sin(O2+90)*cos(O2))*cos(O5))*L6+(cos(O1)*cos(O2+90)*cos(O3)-
cos(O1)*sin(O2+90)*sin(O3))*L3+(-cos(O1)*cos(O2+90)*sin(O3)-
cos(O1)*sin(O2+90)*cos(O2))*L4+cos(O1)*cos(O2+90)*L2

E=

(((sin(O1)*cos(O2+90)*cos(O3)-
sin(O1)*sin(O2+90)*sin(O3))*cos(O4)+cos(O1)*sin(O4))*cos(O5)+(-
sin(O1)*cos(O2+90)*sin(O3)-sin(O1)*sin(O2+90)*cos(O2))*sin(O5))*cos(O6)-

((sin(O1)*cos(O2+90)*cos(O3)-sin(O1)*sin(O2+90)*sin(O3))*sin(O4)-
cos(O1)*cos(O4))*sin(O6)

F=

-(((sin(O1)*cos(O2+90)*cos(O3)-
sin(O1)*sin(O2+90)*sin(O3))*cos(O4)+cos(O1)*sin(O4))*cos(O5)+

(-sin(O1)*cos(O2+90)*sin(O3)-sin(O1)*sin(O2+90)*cos(O2))*sin(O5))*sin(O6)-
((sin(O1)*cos(O2+90)*cos(O3)-sin(O1)*sin(O2+90)*sin(O3))*sin(O4)-
cos(O1)*cos(O4))*cos(O6)

G=
-((sin(O1)*cos(O2+90)*cos(O3)-
sin(O1)*sin(O2+90)*sin(O3))*cos(O4)+cos(O1)*sin(O4))*sin(O5)+(-
sin(O1)*cos(O2+90)*sin(O3)-sin(O1)*sin(O2+90)*cos(O2))*cos(O5)

y=

(-((sin(O1)*cos(O2+90)*cos(O3)-
sin(O1)*sin(O2+90)*sin(O3))*cos(O4)+cos(O1)*sin(O4))*sin(O5)+(-
sin(O1)*cos(O2+90)*sin(O3)-
sin(O1)*sin(O2+90)*cos(O2))*cos(O5))*L6+(sin(O1)*cos(O2+90)*cos(O3)-
sin(O1)*sin(O2+90)*sin(O3))*L3+(-sin(O1)*cos(O2+90)*sin(O3)-
sin(O1)*sin(O2+90)*cos(O2))*L4+sin(O1)*cos(O2+90)*L2

I=
((sin(O2+90)*cos(O3)+cos(O2+90)*sin(O3))*cos(O4)*cos(O5)+(-
sin(O2+90)*sin(O3)+cos(O2+90)*cos(O2))*sin(O5))*cos(O6)-
(sin(O2+90)*cos(O3)+cos(O2+90)*sin(O3))*sin(O4)*sin(O6),

J= -
((sin(O2+90)*cos(O3)+cos(O2+90)*sin(O3))*cos(O4)*cos(O5)+(-
sin(O2+90)*sin(O3)+cos(O2+90)*cos(O2))*sin(O5))*sin(O6)-
(sin(O2+90)*cos(O3)+cos(O2+90)*sin(O3))*sin(O4)*cos(O6),

K=
-(sin(O2+90)*cos(O3)+cos(O2+90)*sin(O3))*cos(O4)*sin(O5)+(-
sin(O2+90)*sin(O3)+cos(O2+90)*cos(O2))*cos(O5),

Z=(-(sin(O2+90)*cos(O3)+cos(O2+90)*sin(O3))*cos(O4)*sin(O5)+(-
sin(O2+90)*sin(O3)+cos(O2+90)*cos(O2))*cos(O5))*L6+(sin(O2+90)*cos(O3)+cos(O
2+90)*sin(O3))*L3+

(-sin (O2+90)*sin (O3)+cos(O2+90)*cos(O2))*L4+sin(O2+90)*L2+L1

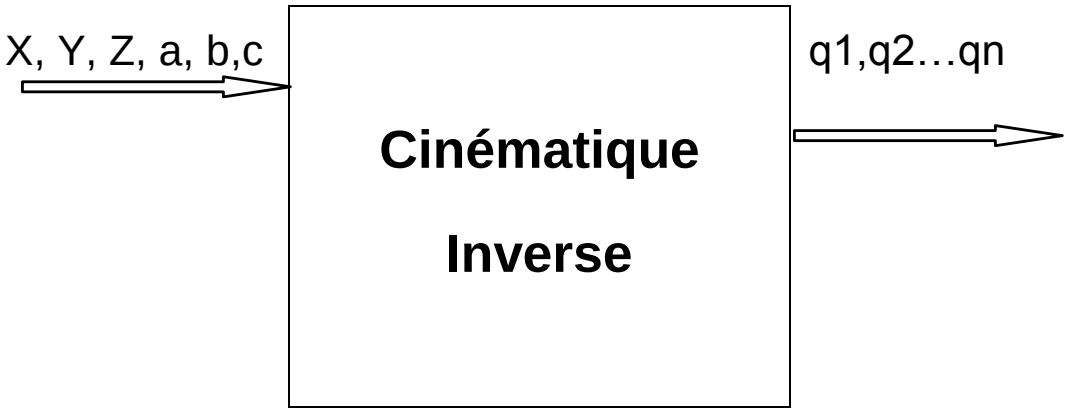
Ainsi, les coordonnées du point du centre de l'outil (TCP) dans le repère absolu sont : **X, Y, Z.**

Problème de la cinématique inverse

Nous avons vu dans le chapitre précédant qu'il est possible de déterminer la position et l'orientation lorsqu'on connait les coordonnes articulaires qi du robot ; en effet, celles-ci correspondent aux valeurs instantanées de θ_i ou de di, suivant qu'il s'agisse d'une rotation ou d4une translation suivant l'axe z_i.

En générale que l'utilisateur recherche plutôt, c'est une méthode permettent de calculer les θ_i ou di pour une position et une orientation déterminée de l'outil.il s'agit la du problème du cinématique inverse :

A partir d'une position et une orientation donnes de l'outil, trouver les coordonnées articulaires du robot.



Ce problème est beaucoup plus complexe que celui de la cinématique directe car pour une position donnée, la solution n'est pas unique (plusieurs configurations possibles).

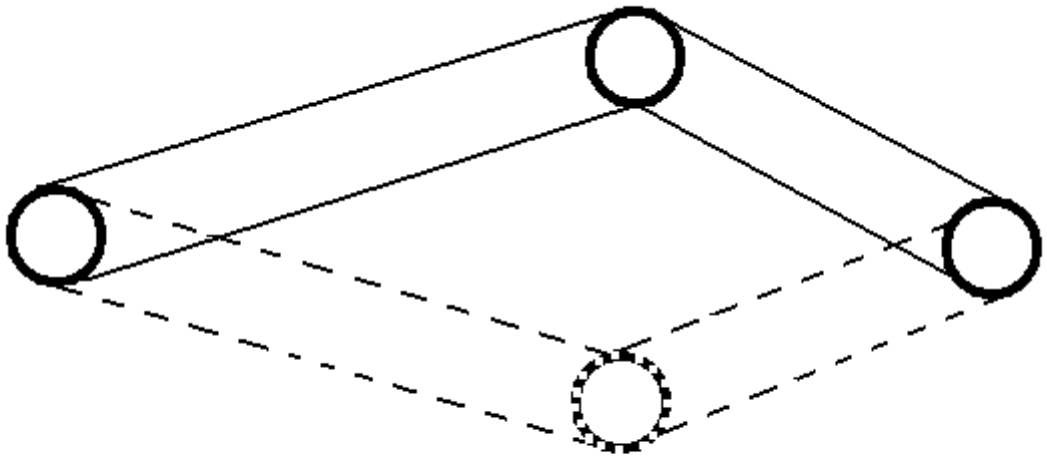


Figure montre la non unicité de la solution.

Pour notre robot le modèle cinématique inverse ne pose pas de problème puisque l'ordinateur nous affiche instantanément tout les coordonnes articulaires a chaque mouvement du robot.

Conclusion.

Le modèle géométrique directe du robot nous permet de trouver les coordon2es opérationnelles en donnant les informations de l'organe terminal en fonction des coordonnes articulaires .Le problème inverse consiste à calculer les coodon2es articulaires correspondant a une location de l'organe terminal.

A travers ce chapitre on acquiert déjà une bonne connaissance sur l'étude théorique du robot.

Il nous reste maintenant d'attaquer la coté pratique pour en savoir utiliser une multitude de commandes sur le robot. Cette cote va être aborde dan le troisième chapitre qui est le plus important pour notre projet.

Chapitre 3 : commande du robot.

Dans le premier chapitre on a présenté le robot Samsung Faraman AS2. Dans le deuxième, on a put le modéliser .Dans ce chapitre on a comme but de maitriser le control du robot par les moyens existant et ensuite on essayera d'intégrer l'intelligence dans les mouvements du robot.

Pour atteindre ce but, on doit tout d'abord, définir l'organe responsable au contrôle direct du robot, les systèmes d'axes et les modes de contrôle. Ensuite, on doit savoir Controller le robot a partir du TP. Puis, il faut bien maitriser le contrôle par le logiciel SRCwin .Enfin on essaye de bouger le robot par C++, ainsi on peut raisonner avant de donner un ordre au robot.

I. Informations utiles pour le control du robot.

1. Le contrôle du robot :

On ce qui concerne le contrôle du robot il existe deux modes opératoires et qui sont les suivants :

- Le mode interne : dans ce mode le contrôle est effectué par le **Teaching Pendant** dans ce mode l'ordinateur communique Via le port série RS232C.
- Le mode externe : dans ce mode, le mécanisme de contrôle est réalisée par un système de signaux digitaux I /O entre le PLC par exemple et le contrôleur.

Les détails de ce système I /O se trouvent dans le manuel d'utilisation.

2. Qui control le robot ?

Le contrôle direct du robot Faraman AS2 est assuré par le SRC (Samsung Robot Controller), il est considéré comme le cerveau de l'installation et équipé d'une unité centrale de traitement d'Intel 80486.

Cette unité est caractérisée par des hautes performances et une haute vitesse d'exécution qui peut contrôler trois robots au même temps.

Le contrôleur est conçu sur la base d'une architecture ouverte, ainsi il peut se connecter par interface à plusieurs genres de robot.

Les caractéristiques facultatives du contrôleur sont une borne, un panneau principal, carte VGA, panneau d'entrée-sortie d'utilisateur, panneau de vision, panneau de sonde de vision de laser, la commande dure de disque, disque souple et ainsi de suite.

FIGURE : LE SRC .

Contrôle par Teaching Pendant.

1. Définition :

Le **TeachingPendant** constitue la principale interface, c'est une sorte de console combiné avec un écran LCD (liquid cristal display), il permet d'accéder aux différentes commandes aussi les exécuter.

Le **Teaching Pendant** peut être utilisé en deux modes :

- Le **jog mode** est un mode de contrôle directe, chaque commande saisie sur **Teaching Pendant** est exécutée directement sur le robot, on enseigne la position au robot puis on l'enregistre.
- Le user mode on doit définir des coordonnés d'avances qui seront atteintes par le robot.

A droite de l'écran LCD, un bouton rouge permet d'arrêter instantanément le mouvement du robot c'est le bouton d'arrêt d'urgence.

A gauche de l'écran LCD se trouve une série de diodes rouges qui renseignent sur l'état général du robot .

FIGURE :LE TP.

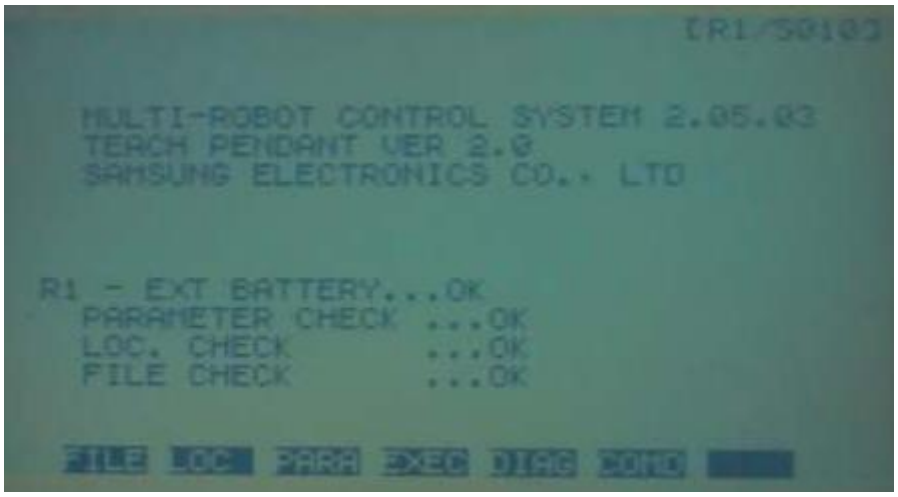
2. **Les signaux émies par les diodes**

- Em_stop : s'allume en cas d'arrêt d'urgence.
- SERVO : s'allume en cas où le moteur est alimenté en énergie.
- H /W LMT : s'allume en cas de limite de matériel.
- S/W LMT : s'allume en cas de limite de logiciel.
- ERROR : s'allume en cas ou il y a problème dans l'exécution.
- RUN : s'allume lors de l'exécution d'un programme.
- STEP : s'allume en cas de fonctionnement d'étape.
- HOLD : s'allume en cas de pause du robot.
- JOINT : s'allume en cas de joint mode.
- WORD : s'allume en cas de world mode.
- TOOL : s'allume en cas de tool mode.
- JOG : s'allume en de jog mode.
- EXT : s'allume en cas de mode externe.
- ZR : s'allume en cas d'un retour au point d'origine.
- ROBOT 2 : indice le robot contrôlé actuellement (FARA-T01).
- ARC ON : s'allume en cas de mode de soudure (FARA-T02).

3. **Les menus principaux relatifs au contrôle du robot.**

Lorsqu' on allume le contrôleur avec l'interrupteur situé a l'arrière, il faut attendre 25 secondes c'est le temps nécessaires pour certaines vérifications.

Apres l'écoulement de ces 25 secondes le TP nous affiche sur son écran les différents menus pour le control du robot.



F1	F2	F3	F4	F5	F6	F7

FIGURE : premier affichage du TP.

- ❖ FILE : contrôle le programme utilisateur. lorsque on appuis sur la touche ' F1', la clef vous porte à l'écran de gestion de dossier.

- LOC : l’appui sur la touche ‘F2’ vous montre les coordonnés enregistrées par le robot (les coordonnées vert les quels le robot pourrait être amener à se rendre.
- PARA : en appuyant sur la touche ‘F2’ on peut gérer les paramètres concernant le corps du robot (tel que la vitesse des moteur, les angles de rotation .. .) et ceux concernant les signaux digitaux I /O entre la pince et le contrôleur.
- EXEC : va à l'écran où le langage du robot peut être exécuté.
- DIAG : fait du diagnostique dans les divers opérations du robot (par exemple les erreurs survenues et les positions actuelles du robot), spécifications et arrangement de port d'entrée-sortie.
- COMD : c’est pour accéder à certaines commandes spéciales de l’utilisateur.

4. **Les fonctions du Teaching Pendant :**

ZR : fait le retour à zéro.

SERVO ON /OFF : ouvrir et fermer les cerveaux des moteurs.

MODE : changer les systèmes de coordonnées.

JOG : permet de changer entre le jog mode et user mode.

SP : met un espace lors de l’écriture d’un programme ou entrer une commande.

REC : enregistrer la position actuelle.

PAUSE : arrêter et reprendre le mouvement en cas d’une opération.

ENTER : exécute la fonction écrite dans le menu.

SPEED : augmenter ou diminuer la vitesse du déplacement, sélectionner le robot .

ROBOT :

DEFGERTRTRRUN : utilisé comme clé de basculant, lire / arrêter le robot.

FIGURE : Les boutons du TP.

1. Les boutons du menu.
2. Bouton de la deuxième fonction.
3. Les boutons de direction.
4. Les boutons de jog.

5. **Comment bouger le robot avec le TP ?**

Pour faire bouger le robot avec le **Teaching Pendant** il suffit de :

- Allumer le **SRC**.
- Choisir le menu **EXEC** en appuyant sur **F4**.

Plusieurs locations déjà enregistrées et Cinq commandes comme **MOVE**, **MOVA**, **REDY, MOVS** et **UP** s'affichent sur l'écran.

- Choisir **MOVE** en appuyant sur **F1**.
- choisir la location voulue avec la touche entrer.

Ainsi il s'affiche sur l'écran la commande et la location.

- Pour exécuter il faut au même temps appuyer sur les touches **SHIFT** et **ENTRER.**

Comment contrôler le robot à partir d'un logiciel sur ordinateur :

La deuxième méthode qu'on peut utiliser pour dialoguer avec le contrôleur c'est via le logiciel **SRCwin** installé sur ordinateur, cette méthode est plus facile à utiliser car elle nous offre un écran plus grand et plus facile a comprendre mais le temps d'exécution par cette méthode est supérieur au temps mis par le TP à exécuter une tâche .

1. Lancer le logiciel et se connecter avec le SRC.

Pour utiliser ce logiciel il faut suivre les étapes suivantes :

- Lancer le logiciel de commande du robot **SRCwin** en cliquant sur l'icône **SRCwin.exe**.

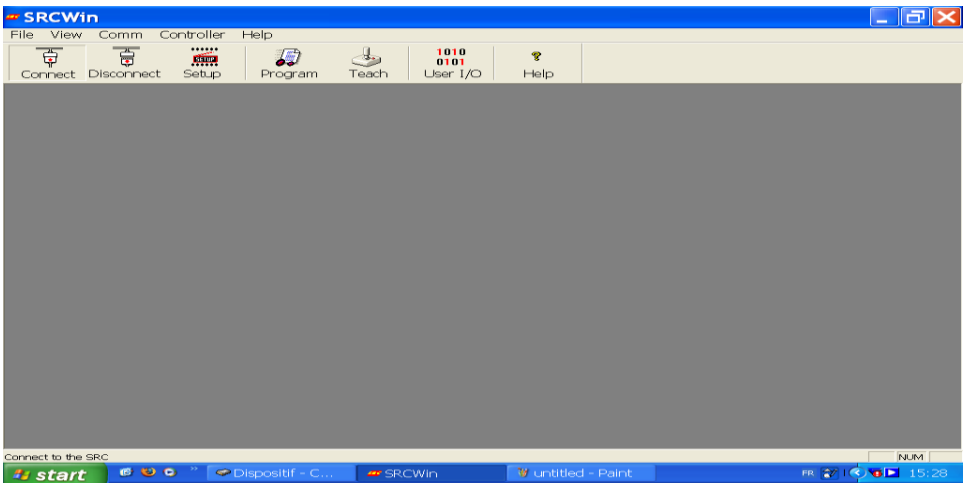


FIGURE : premier affichage de SRCwin.

- Choisir les paramètres du menu **Setup:**

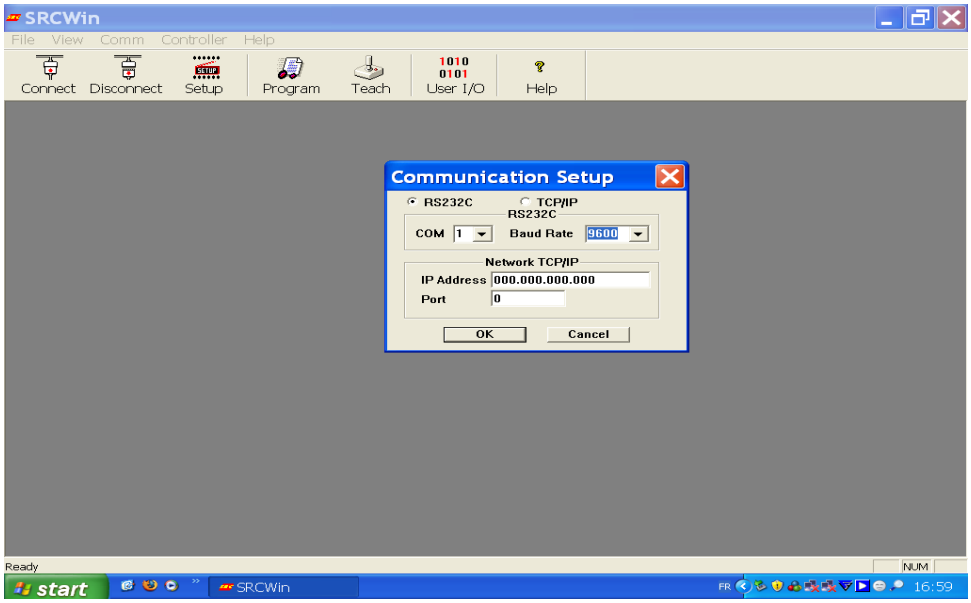


FIGURE : Menu setup.

- Cliquer sur **Connect** pour ouvrir le dialogue entre le logiciel et le contrôleur du robot à travers le port série **SRC232** (COM 1 ou COM 2).

Il faut que les indicateurs à droite **TRSF** et **RECV** s'allument alternativement pour nous informer que l'émission et la réception des informations ont commencé et qu'on peut commencer à commander le robot.si non, on lance de nouveau le logiciel et on répète les même étapes jusqu'à obtenir un dialogue entre le logiciel et le contrôleur. En ce moment, le contrôleur et le logiciel de commande sont connectés.

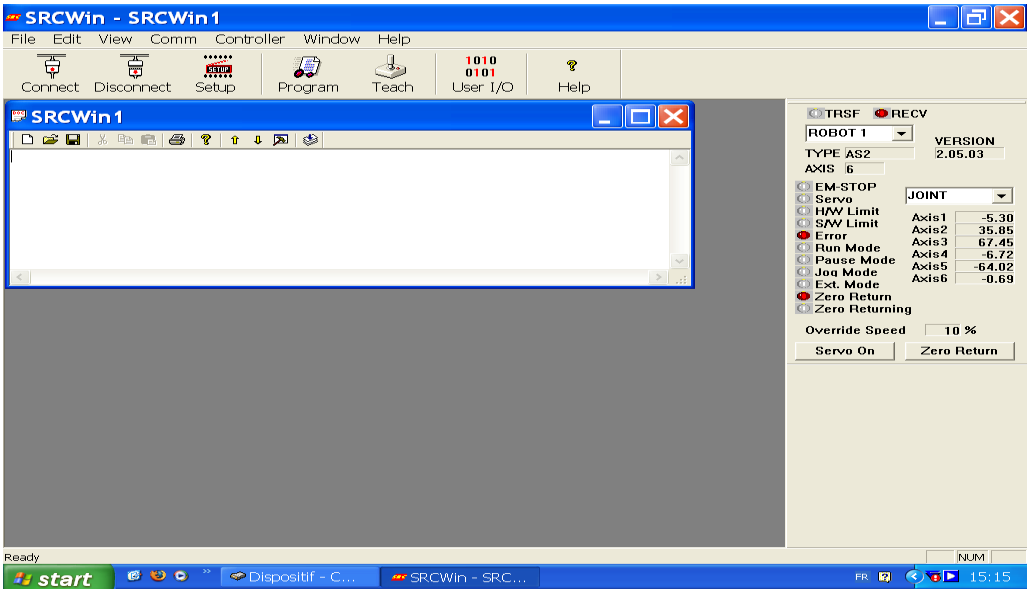


FIGURE : comment connecter.

- Cliquer sur **Servo on** pour ouvrir les cerveaux moteurs du robot.

- Cliquer sur **Zero Return** pour initialiser la position zéro du robot.

2. Choisir le menu de notre besoin

Maintenant on peut choisir entre plusieurs menus.

a. Menu Program :

Le langage de programmation est l'outil principal de communication avec le contrôleur de robot (SRC), il fournit diverses capacités telles que :

- créer un programme ou le modifier

- contrôlant les données des positions enregistrées

- exécutant un programme et etc……

Comme avec tous les robots de Samsung, l'as2 est programmé utilisant le langage de programmation de FARAL

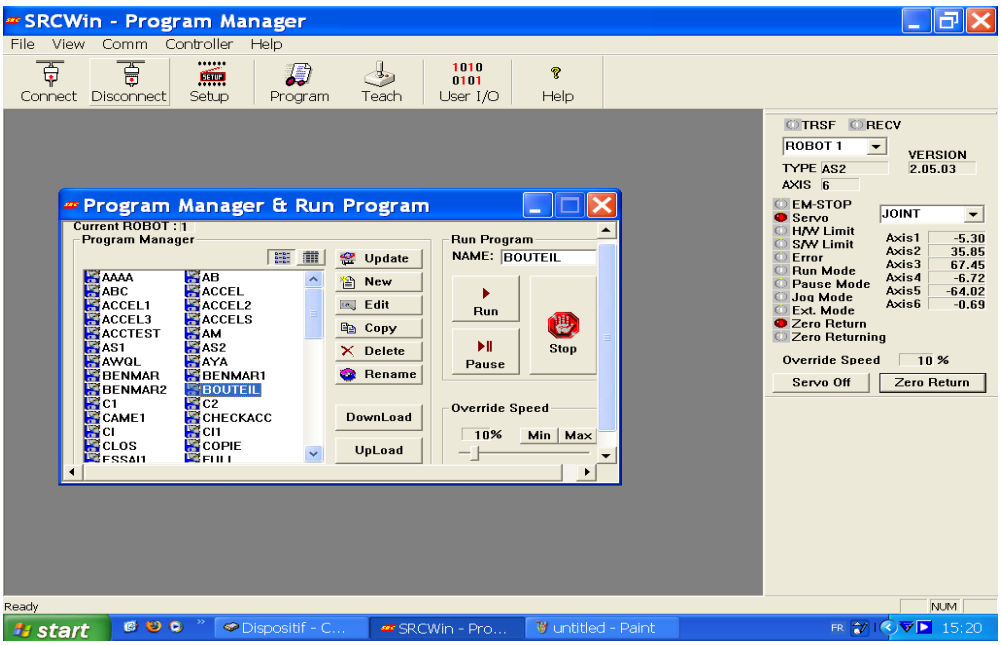


FIGURE : Menu Program.

Pour créer un nouveau programme on doit suivre les étapes suivantes :

1. Cliquer sur **Programme**.
2. Attendre la réception des noms de tous les programmes.
3. Cliquer sur **New**.
4. Ecrire le programme, compiler et enregistrer.

Comment exécuter un programme ?

Lors de l'exécution d'un nouveau programme, il est prudent de s'assurer d'une vitesse relativement faible pour des raisons évidentes de sécurité (par exemple 10% de la vitesse maximale).

1. Télécharger ce programme en cliquant sur **DownLoad**.
2. Choisir le programme à exécuter.
3. Cliquer sur **Run**.

Quelques commandes pour faire bouger le robot :

INT A, B : déclarer les entiers.

REAL C, D : declarer les reels.

SIG 1 : fermer la pince.

SIG -1 ouvrir la pince.

DELAY 1 : arrêter l'exécution du programme pendant 1 seconde**.**

MOVE #ZERO : aller à la location #ZERO.

END : essentiel à la fin de chaque programme.

LOC P1,#P2 : déclaration des locations.

GOTO 40 : sauter à la commande désignée par 50

LSET P1 =200, 130, 200,-80, 45,-90 : mettre les locations dans la location P1.

LCOPY P1 P2 : copier les donnes de P1 dans P2.

STOP : arrêter l'exécution.

Draw 10, 0, 0 : bouger 10 mm dans la direction des X et avec le système des coordonnées cartésien.

DRIVE 1, 80, 40 : tourner l'axe 1 de 80° avec 40 % de vitesse.

UP 30 : Se déplacer vers le haut jusqu'à la position où L'axe Z est de 30 mm.

ROTATE 90 : faire tourner le robot suivant l'axe des z dans le système de coordonnées outil par 90°.

Exemple de programme écrit sur ordinateur pour faire bouger le robot:

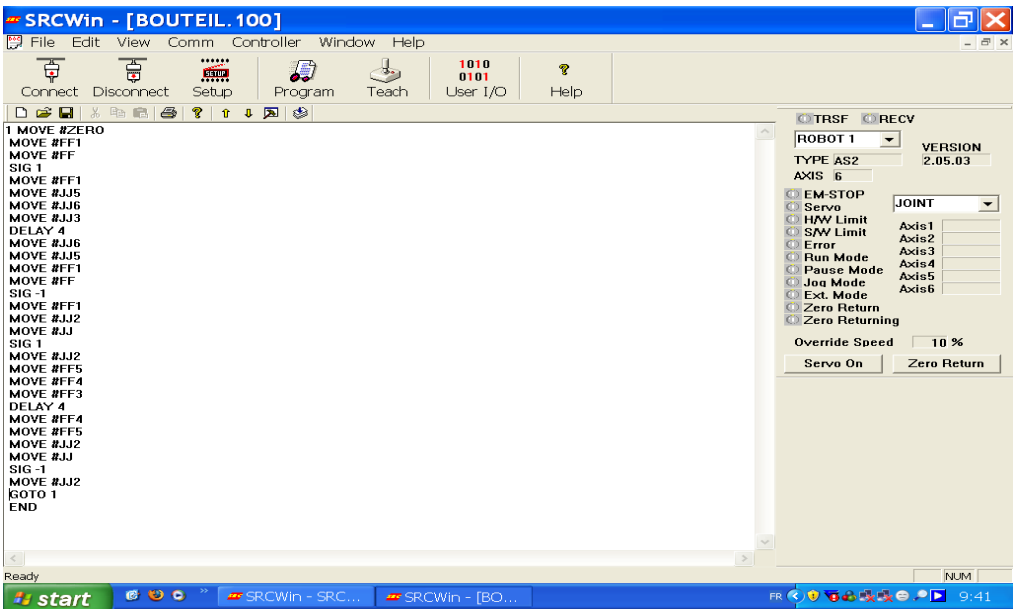


FIGURE : Exemple de programme.

Menu Teach :

On peut contrôler le robot de la même manière que le **TeachPendan**t, pour cela, il suffit de suivre les étapes suivantes:

1. Cliquer sur **Teach.**
2. Attendre un moment, le temps que les informations sur l'état du robot et ces différentes locations soient reçues.

On peut choisir entre **jog mode** et **user mode**.

Si on veut commander le robot directement, on coche le case **JogEnable (fig)** pour passer au mode **JOG** et la on peut piloter le robot par les flèches vert comme si on commande avec le **TP**.

Si on veut enregistrer une position, on peut enregistrer toute position voulue en choisissant un nom puis en appuyant sur **Recorde,** pour cela, on choisir le système des coordonnés qui convient le mieux.

Selon le système d'axes utilisé, on a certaines règles à suivre :

*Le nom est limité à sept caractères alphanumériques.

*Les locations en coordonnées cartésiens commencent toujours par des lettres tandis que pour les locations en coordonnées des articulations ces lettres sont précédées du signe#.

FIGURE : Menu Teach avec JogEnable.

- Pour donner au robot des ordres pour atteindre des coordonnés enregistrées il faut que :

1. **JogEnable** ne soit pas cochée **(fig)**.
2. on écrit **Move** dans le premier espace.
3. Dans le second espace, on écrit le nom de l'emplacement voulue, ou il suffit de choisir l'un des coordonnées enregistrées.
4. On choisi la vitesse désirée en pourcent.
5. on clique sur **EXECUTE.**

-Pour controller la pince il faut que:

1. **JogEnable** ne soit pas cochée **(fig)**.
2. on tape **SIG 1** pour l'ouverture ou **SIG -1** pour la fermeture dans le premier espace.
3. dans le second espace on n'écrit rien.
4. on clique sur **EXECUTE.**

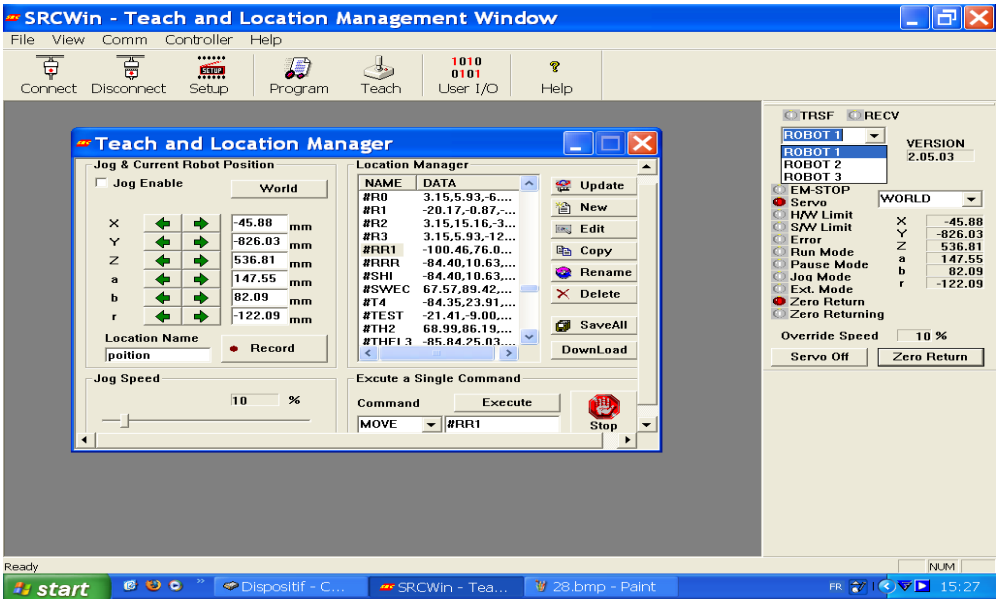


FIGURE : Menu Teach sans JogEnable.

Ce programme peut contrôler 3 robots au même temps comme l'indique la figure en dessus.

A droite et en haut, on trouve les étapes qui indiquent l'état du robot comme celle du **Teaching Pendant.**

Menu User I/O :

Pour travailler en mode externe un système des signaux O/I est utilisé pour commander les différents outils pouvant être montés au poignet. Ce système contient des commandes qui peuvent être intégrée dans la programmation du robot.

Pour commander la pince en fermeture, il suffit de cocher la première case de **Output** (fig) et pour l'ouvrir, il suffit de l'enlever.

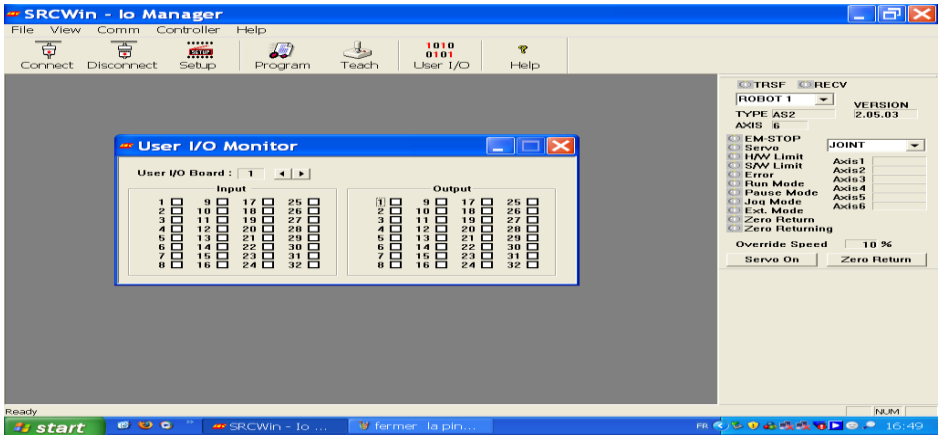


FIGURE : Menu User I /0 avec pince ouverte.

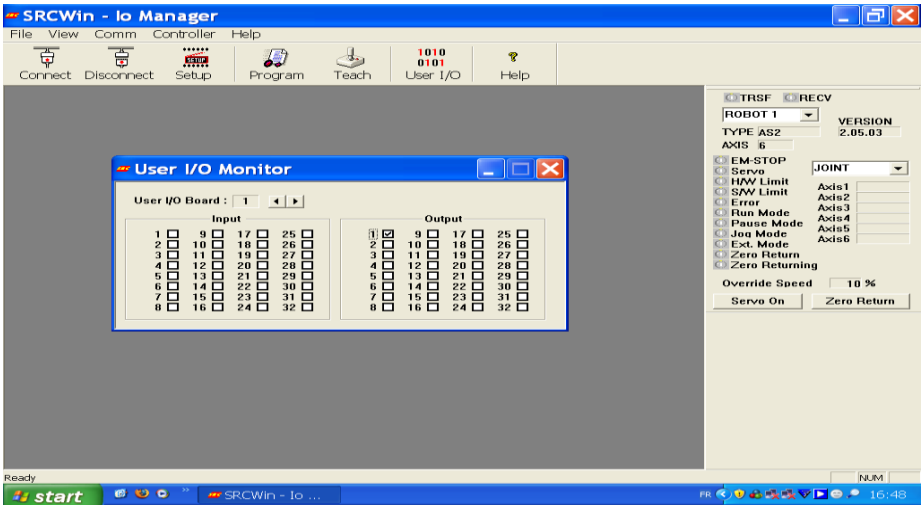


FIGURE : Menu User I /0 avec pince fermée.

Contrôler le robot à partir de C++ :

Les deux méthodes évoquées précédemment sont utilisées pour un contrôle bête du robot, on donne au robot une ou une succession d'ordres et il les exécute sans réfléchir ni raisonner.

Mais pour jouer à l'échec, il faut raisonner puis ordonner un mouvement bien défini au robot.

Donc il faut commander le robot par un logiciel qui peut en temps réel raisonner en suivant l'état du jeu, réfléchir à la meilleure solution et donner une succession d'ordres au robot.

Et pour cela il faut :

1. Trouver le meilleur logiciel pour analyser le dialogue entre l'ordinateur et le contrôleur à travers le port série RS232

2. Faire bouger le robot avec le logiciel sur ordinateur SRCwin et analyser ce dialogue avec le logiciel d'écoute sur port série.
3. Ecrire un programme C++ qui peut envoyer des donnes sur le port série.
4. Essayer de faire bouger le robot en envoyant un même message que celui envoyé par SRCwin envoie pour faire bouger le robot.
5. Faire la même démarche pour Controller la pince.

1. La recherche d'un programme qui détecte le dialogue entre l'ordinateur et le contrôleur :

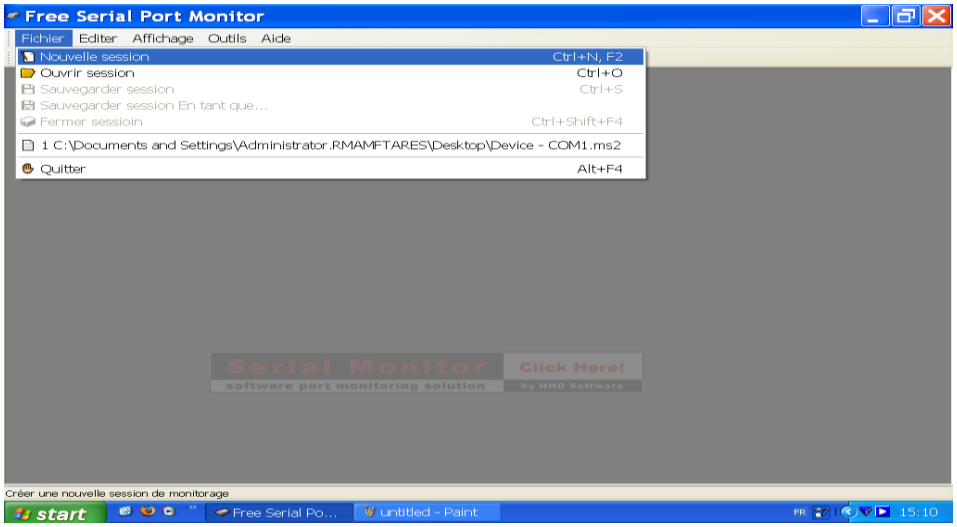
Une recherche sur internet des programmes d'écoute sur le port série m'a permis de télécharger plusieurs programmes tel que **SHDCom**, Devise **monitoring studio** et **HHD free-serial-port-monitor**.

Pour mon travail, j'ai choisi ce dernier pour plus de souplesse d'utilisation : il vous offre plus de choix et un affichage facile a comprendre.

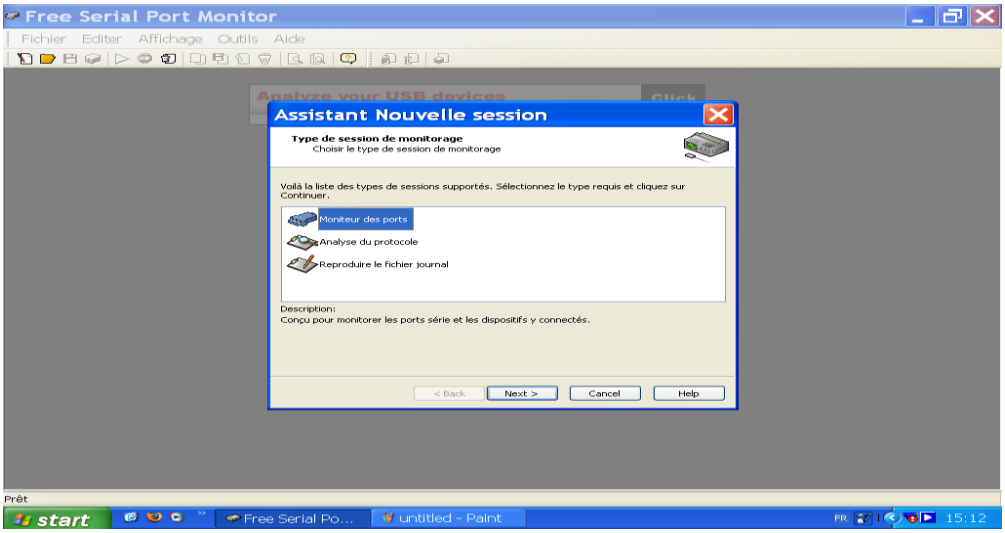
2. Comment utiliser le logiciel HHD free-serial-port-monitor ?

Pour utiliser ce programme il faut :

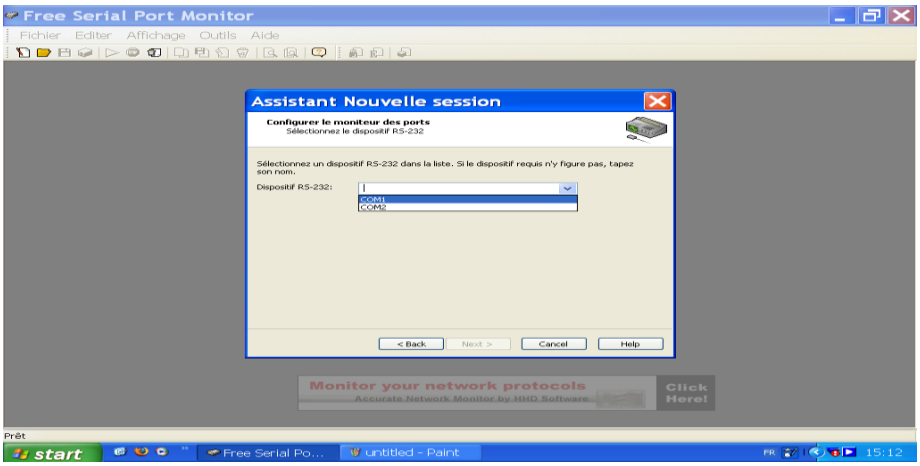
- Lancer et ouvrir une nouvelle fenêtre de travail.



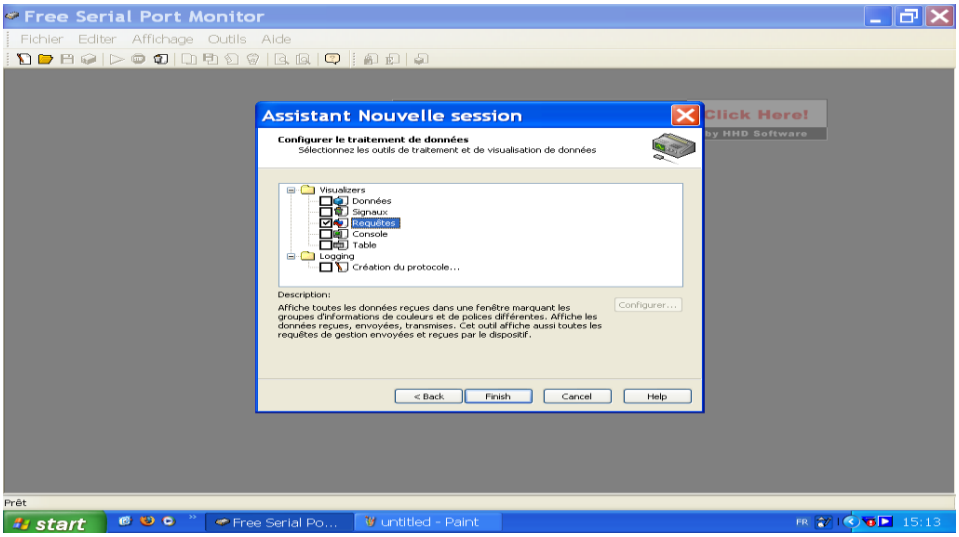
- Choisir **Moniteur des ports.**
- Cliquer sur **Next**.


Free Serial Port Monitor
Fichier Editer Affichage Outils Aide
Assistant Nouvelle session
Type de session de monitorage
Choisir le type de session de monitorage
Voilà la liste des types de sessions supportés. Sélectionnez le type requis et cliquez sur Continuer.
Moniteur des ports
Analyse du protocole
Reproduire le fichier journal
Description:
Conçu pour monitorer les ports série et les dispositifs y connectés.
< Back
Next >
Cancel
Help
Prêt
start
Free Serial Po...
untitled - Paint
15:12

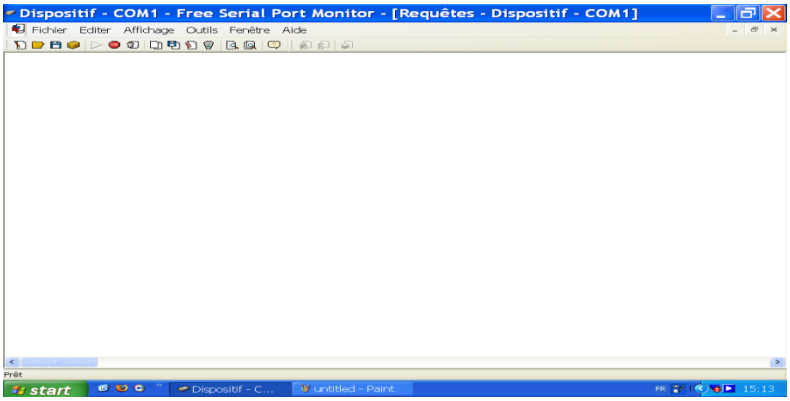
- Choisir le numéro du port qu'on va utiliser (COM1 ou COM2).



- Cocher la case **Requêtes** et cliquer sur **Finish.**



- Une page de travail vierge s'affiche .


Dispositif - COM1 - Free Serial Port Monitor - [Requêtes - Dispositif - COM1]
Fichier Editer Affichage Outils Fenêtre Aide
Prêt
start
Dispositif - C...
untitled - Paint
15:13

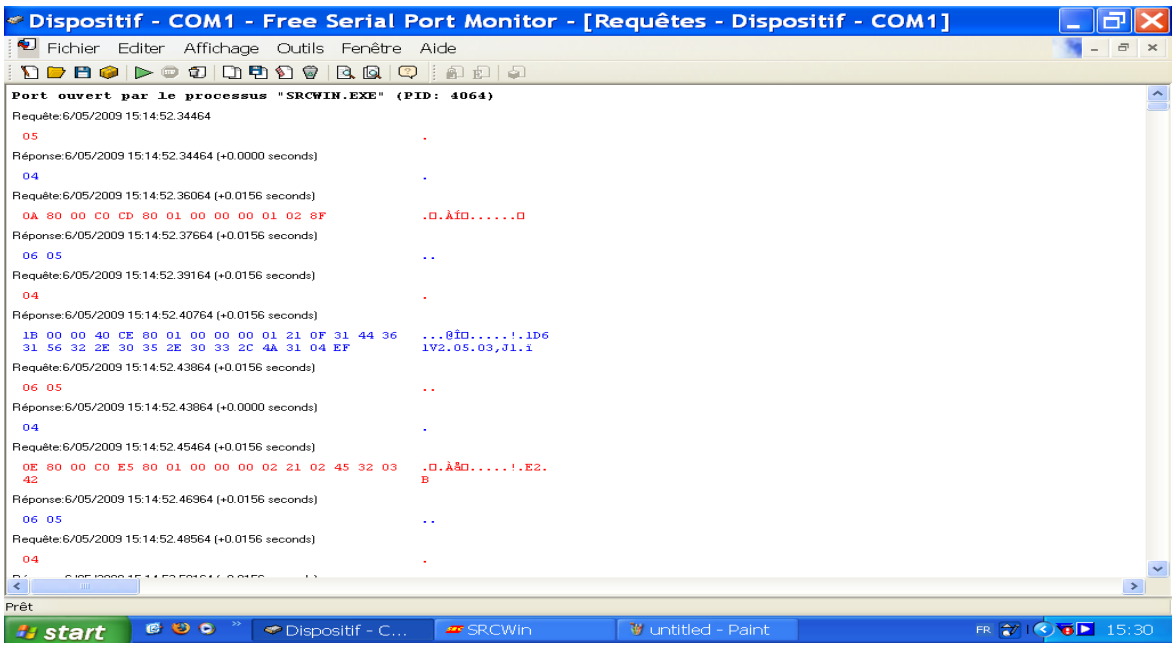
- Démarrer le programme qu'on veut écouter sur le port série.

3. **Remarque :**

Si on ouvre le port série et on lance **HHD free-serial-port-monitor,** le port série qui a été ouvert ne s'affichera pas et on ne pourra pas le lire, donc il faut tour d'abord, ouvrir **HHD free-serial-port-monitor** puis commencer à envoyer les messages sur le port série.

Dés qu'on ouvre le port série on peut différentier entre les requêtes envoyées et reçues, les requêtes émises sont en rouge et celles reçues sont en bleu.

On peut enregistrer ces requêtes pour l'analyser tranquillement.



Bouger le robot avec SRCwin et analyser le dialogue entre l'ordinateur et le contrôleur Avec le logiciel d'écoute **HHD free-serial-port-monitor.**

On a besoin de détecter le signal émis par l'ordinateur pour faire bouger le robot.

1. On ouvre **HHD free-serial-port-monitor**
2. Puis on ouvre **SRCwin**,
3. On commande le robot de bouger pour une location en effectuant la commande ' **move #zeroo'.**

⇨ On remarque que cet ordre a été détecté, écrit et exécuté sur le port comme c'est indiqué sur la figure suivante.

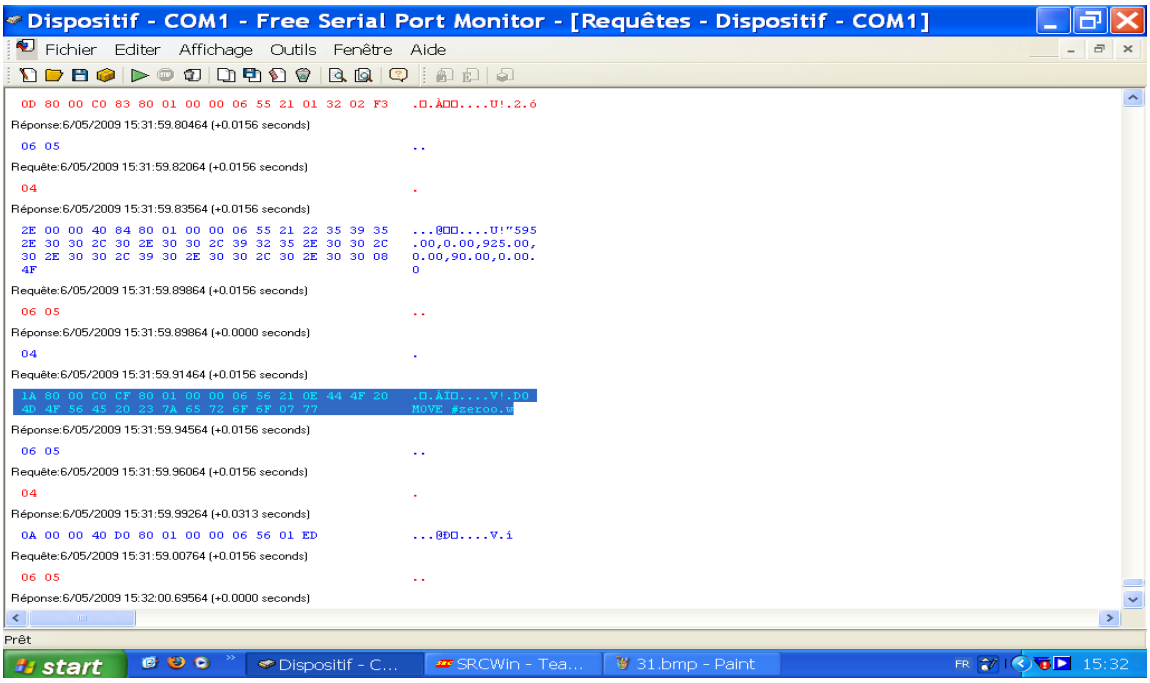

Dispositif - COM1 - Free Serial Port Monitor - [Requêtes - Dispositif - COM1]
Fichier Editer Affichage Outils Fenêtre Aide
Réponse:6/05/2009 15:31:59.80464 (+0.0156 seconds)
Requête:6/05/2009 15:31:59.82064 (+0.0156 seconds)
Réponse:6/05/2009 15:31:59.83564 (+0.0156 seconds)
Requête:6/05/2009 15:31:59.89864 (+0.0156 seconds)
Réponse:6/05/2009 15:31:59.89864 (+0.0000 seconds)
Requête:6/05/2009 15:31:59.91464 (+0.0156 seconds)
Réponse:6/05/2009 15:31:59.94564 (+0.0156 seconds)
Requête:6/05/2009 15:31:59.96064 (+0.0156 seconds)
Réponse:6/05/2009 15:31:59.99264 (+0.0313 seconds)
Requête:6/05/2009 15:31:59.00764 (+0.0156 seconds)
Réponse:6/05/2009 15:32:00.69564 (+0.0000 seconds)
Prêt
start
Dispositif - C...
SRCWin - Tea...
31.bmp - Paint
15:32

FIGURE : Détection de la commande pour bouger le robot.

Maintenant il faut essayer d'envoyer tout le message sélectionné dans la figure via c++, d'où la nécessité de trouver un programme C++ qui envoie les messages dans le port série.

Controler le robot avec c++.

1. Installation du port série

on a utilisé le code du **FoggyLog [23]** pour sa simplicité d'utilisation (moins de commandes à faire). Ce programme ouvre le port série, envoie

un message et il le referme le port série, sinon il vous indique un message en cas d'erreur

le manipulateur est connecté au COM1 et communique avec un baudrate de 9600

Pour tester l'envoi du caractère sur le port série, il suffit d'ouvrir le logiciel d'écoute **HHD free-serial-port-monitor**, puis exécuter le programme de commande de port série.

On remarque que le **HHD free-serial-port-monitor** a détecté '48', qui est '72 'en hexadécimal. Donc on peut conclure que les informations qui passent dans le port série sont en hexadécimal.

Il y a des exceptions lors du codage du hexadécimal en décimale telle que:

Tout nombre qui commence par zéro et y à droite un chiffre on lui ajoute le nombre « un » à gauche.

Exemple :

Si on trouve 0E elle devient 10E puis on procède au codage de ce dernier en décimal.

2. **Essai 1.**

a. **Description**

Pour faire bouger le robot il faut tout d'abord ouvrir le logiciel d'écoute. Ensuite, faire bouger le robot avec SRCwin , détecter la chaine qui donne l'ordre de bouger puis, convertir ces chaines en décimal. Enfin, écrire cette chaine avec c++ et essayer de l'envoyer pour faire bouger le robot.

#h11 est une position déjà enregistrée dans le contrôleur, on a voulue faire bouger le robot avec cette location, décodant ce que **HHD free-serial-port-monitor** détecte et copier cette chaine et l'envoyer avec c++.

Lorsqu'on bouge le robot avec SRCwin le logiciel d'écoute nous a donne cette chaine :

'18 80 00 C0 CF 80 01 00 00 02 01 21 0C 44 4F 20 4D 4F 56 45 20 23 68 31 31 05 B7 '

Qui est la même que la suivante **:**

'.□.ÀÏ□.....!.DO MOVE #h11. ·'

On décode toute cette chaine de chiffres en décimal et on essaye d'envoyer tout avec c++.

Résultat :

Le logiciel **HHD free-serial-port-monitor** a écouté la même chaine que le SRCwin à envoyée. Mais le robot ne bouge pas.

3. Essai 2.

a. Description

→La meilleur solution c'est d'émettre toutes les requêtes que le SRCwin à envoyée dès qu'on connecte et jusqu'a l'envoi du l'ordre de faire bouger pour **#h11**.

Pour cela, il faut décoder toutes les requêtes et les envoyer successivement dans le port série.

Résultat :

Le logiciel **HHD free-serial-port-monitor** a détecté la même chaine que le SRCwin a envoyée et le robot bouge vers la location **#h11**.

Remarque :

le robot ne bouger que s'il effectue une série de conversations avec l'ordinateur, exactement on envoie dis requêtes et on reçoit dis ; après ce dialogue de dis lignes on peut envoyer notre ordre.

Ces requêtes de dialogue sont pour dire que le robot est connecté et qu'il est prêt à recevoir des ordres.

Donc ces dis lignes de dialogue sont nécessaires avant de donner tout ordre au robot.

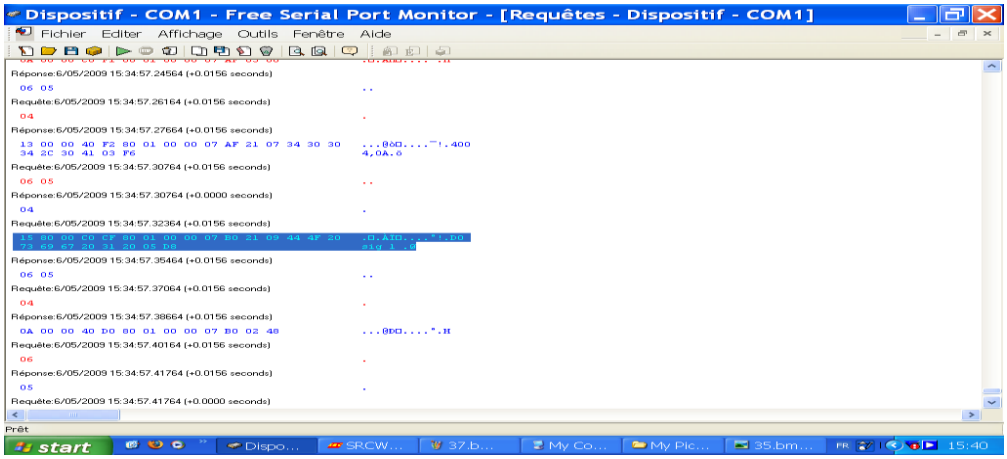
Pour faire ouvrir et fermer la pince, on doit écouter ce que SRCwin envoi dans le port série, ensuite, on refait le même travail fait pour bouger vers une location.

Ouvrir et fermer la pince avec c++.

Apres avoir décodé le message, on envoie avec c++ les dis requêtes de dialogue suivies par la requête de l'ordre pour ouvrir la pince' **sig 1'.**

On voie dans la figure () que le logiciel detecte cette commande et la pince se ferme.

On fait le même travail avec **sig -1**, la pince se ferme.


Dispositif - COM1 - Free Serial Port Monitor - [Requêtes - Dispositif - COM1]
Fichier Editer Affichage Outils Fenêtre Aide
Prêt
start

Conclusion :

Pour donner un ordre au robot il faut suivre les étapes suivantes:

+ Si cet ordre sera une location il faut l'enregistrer avec **SRCwin**.

+ Écouter cet ordre avec **HHD free-serial-port-monitor**.

+ Décoder le message en décimal.

+ Envoyer les dis requêtes de dialogue suivi par l'ordre voulue.

Chapitre quatre : Le moteur d'échecs.

Apres avoir parvenu à faire bouger le robot avec le programme c++, on va essayer dans ce chapitre d'introduire un programme de jeu d'échecs au robot. Donc on aura besoin d'un moteur de jeux d'échec avec un code source accessible en c++.

I. Les moteurs de jeu d'échecs

1. Définition

Le moteur d'échecs est un programme informatique d'échecs utilisant un protocole de communication qui permet de dialoguer avec une interface utilisateur.Ces moteurs ne possèdent pas leurs propres interfaces à eux mais ils fonctionnent uniquement par des interfaces comme : XBoard /WinBoard, Arena, ou chessbase (commercial). Ces interfaces permettent également de confronter les moteurs entre eux.

2. Approche historique

La première machine joueuse d'échecs fut l'automate turc (autour de 1789), qui joua notamment contre la tsarine Catherine II et Napoléon Bonaparte, c'était en fait un homme qui se cachait à l'intérieur dans la machine.

Ce n'est que dans les années 1950 (qui coïncident avec le développement des ordinateurs) que les véritables programmes virent le jour. Et ca n'a demandé que 50 années pour que ces programmes passent de la connaissance des règles à un niveau au moins égal à celui des meilleurs joueurs humains.

En mai 1952, Alan Turing (un des pionniers) a écrit un programme de jeu d'échecs. Il simule lui-même les calculs de la machine car il ne dispose pas d'ordinateur assez puissant pour le faire tourner, et il met environ une demi-heure pour effectuer chaque coup. Ce programme perdit contre Alick Glennie, un collègue de Turing.

3. Actualités

Parmi les meilleurs programmes d'échecs figurent actuellement, Rybka, Fritz, Shredder, Hydra ou encore Crafty (un moteur dont on peut accéder au code source). The Chess Network est l'un des projets de calcul reparti qui utilise la puissance de plusieurs calculateurs .Ce projet a regroupé simultanément 2070 calculateur réalisant ainsi en 2004 le record du nombre d'ordinateurs autour d'une partie d'échecs. Il existe également GNUChess.Avec l'evolution de l'ordinateur, ces programmes se sont développés et ils possèdent aujourd'hui leurs propres interfaces (ils sont souvent en licence propriétaire) et d'autres peuvent être utilisés avec l'interface de sont choix et on peut même avoir plusieurs programmes dans une interface.

Récemment le programme Deep Fritz (qui n'est pas le plus puissant des programmes) a battus en 2006 le champion du monde d'échecs le Russe Valdimir kramnik.

Les trois meilleurs programmes sont certainement Rybka 1.0, Fruit 2.3.1 et Toga 1.3.1. Ils ont un niveau de jeu au-delà de 2800 elo. Aujourd'hui il existe de nombreux programmes d'échecs gratuits.

4. interface

La plupart des moteurs d'échecs ne disposent pas de leur propre interface utilisateur graphique (GUI), mais sont plutôt des applications de console pour communiquer avec une interface graphique telle que XBoard (Linux) et WinBoard (Windows) par l'intermédiaire d'un protocole standard.

5. Grand Tournoi Moteurs d'échecs

Le Grand Tournoi des Moteurs d'échecs, aussi connu comme CEGT, est l'une des organisations les plus connues pour les tests de logiciels d'échecs en jouant aux programmes d'échecs les uns contre les autres et de produire un tableau de cotés.

CEGT régule des tests amateurs et professionnels d'échecs dans les différents moteurs de temps tels que les formats 40 / 4 (40 coups en 4 minutes, de répéter), 40/40 (40 coups en 40 minutes, de répéter), et 40/120 (40 coups en 120 minutes , la répétition). Les résultats 40/120 sont considérés les meilleurs jeux d'échecs gratuit par ordinateur disponible en ligne.

6. championnat du monde d'échecs des ordinateurs

Chaque année, on organise le championnat du monde d'échecs des ordinateurs (World Computer Chess Championship, WCCC) qui est une compétition annuelle entre les programmes d'échecs. Elle se déroule souvent en parallèle avec les olympiades informatiques,(des tournois entre ordinateurs pour d'autres jeux).
les programmes d'échecs de cette compétition son basés sur des superordinateurs, des microprocesseurs ou du matériel dédié au jeu d'échecs.

7. championnat du monde des micro-ordinateurs(WMCCC)

Des cycles séparés de championnat réservés aux micro-ordinateurs se sont tenus de 1980 à 2001. Les vainqueurs en étaient les ordinateurs dédiés, mais depuis 1991, les programmes tournant sur des ordinateurs personnels sont eux qui ont dominé la compétition.

Le programme de Richard Lang détient le record avec :

- ✓ 7titres (de 1984 à1990).
- ✓ 10 titres pour les programmes commerciaux (1984 à1993).

Ce championnat est organisé par l'ICGA (International Computer Games Association)

Comment les moteurs d’échecs fonctionnent.

Le jeu d’échecs serait un jeu d'information parfaite, signifiant que toute l'information est évidente sur les deux côtés à tout moment. Ceci permet à un ordinateur de jouer aux échecs en recherchant l'arbre de jeu pour des meilleurs mouvements possible. Ainsi un moteur simple d'échecs se compose de trois parties qui définissent comment elles jouent :

- Un générateur de mouvement
- Une fonction d'évaluation
- Une fonction de recherche

Les mêmes composants de base sont présents dans n'importe quel jeu d'information parfaite, telle que le jeu de dames.

1. Composition du jeu d’echecs

a. Générateur de mouvement

Un générateur de mouvement est capable de produire tous les mouvements légaux d'échecs donnant une interface particulière de la planche. Cette fonction est nécessaire par la recherche pour produire des mouvements afin de passer à travers l'arbre de jeu. C’est critique que la présente partie du moteur d'échecs soit très rapide car elle va être appelée peut-être des millions des fois pour un simple mouvement.

Il y a aussi beaucoup bien d'information à prendre en considération dans la cette partie, comme l'information retranchée, des captures, en passant, la règle de 50 mouvements, des sauts de double et des promotions. Dans TSCP (Tom Kerrigan's Simple Chess Program), Tom utilise un petit peu des composants dans sa structure de mouvement pour stocker une information pareille.

La fonction d'évaluation

La fonction d'évaluation est exactement à ce qu’elle parait être. Elle regarde un état de planche et lui donne des points - typiquement positifs si la planche favorise le blanc et les négatifs pour le noir. C'est essentiellement la source de connaissance d'échecs pour votre moteur. Des fonctions d'évaluation simples peuvent être construites en correspondant les valeurs de pièces pour chaque côté, c.-à-d. 9 pour la reine, 5 pour la tour, 3 pour fou et cavalier et 1 pour le pion. Des fonctions d'évaluation plus complexes contiennent l'information de position telle que le nombre de places attaquées par côté, la commande du centre, doublée de pion vers le haut. Il est également impératif que vous rendez la fonction d'évaluation aussi rapide que possible, pour les mêmes raisons que le générateur de mouvement.

La fonction de recherche

Comme j'avais dit avant, la fonction de recherche cherche l'arbre de jeu et renvoie le meilleur mouvement possible. Comment fait-elle ceci ? Elle utilise le générateur de

mouvement pour produire touts les mouvements possibles de la position puis elle appelle périodiquement la fonction de recherche sur chacune des positions.

Cet algorithme s'appelle le minimax (voir ci-dessous). La raison de son appellation le minimax est qu'il essaye de maximiser les points possibles pour le blanc à une seule intensité de recherche et en réduisant au minimum les points pour le blanc au prochain. Ceci nous permet de saisir le mouvement qui mène aux meilleurs points pour le blanc (ou le noir) à la fin. C'est fondamentalement une force brutale pour chercher toutes les meilleures positions possibles. Maintenant ce n'est pas très efficace, non? Il y a beaucoup d'algorithmes de recherche plus rapides ailleurs. Je vais citer juste quelques un comme référence :

• L'élagage d'Alpha bêta, qui nous permet de saisir la même variation principale qu'avec l'algorithme de recherche de minimax. Il fait donc d'une façon beaucoup plus rapide en présentant le concept d'élagage dans ceci si certains points sont absolument terribles après une recherche, il cause une coupure qui arrête la recherche de cette branche de l'arbre. Ça dépend fortement de la commande de mouvement. Dans le meilleur cas, il tient compte d'une accélération de racine carrée comparé au minimax, et dans le pire de cas, il est juste bon comme le minimax.

• Negascout, qui a été démontré pour accélérer des recherches d'échecs d'environ 10%. C'est essentiellement pareil que la recherche de variation principale de tous les buts pratiques.

• MDT (f) - il s'avère qu'il va être rapide, mais dépend fortement de la transposition des tables

• SSS* - dépend également de la transposition des tables.

• Aspiration Windows, qui tient compte d'un élagage plus sélectif de l'arbre de recherche. Il y a un sort entier de plus que celui que je veux l'ajouter ici : Les transpositions tables, les recherche de repos, etc., mais celui devraient être bons pour maintenant. Je vous propose fortement de vérifier ma page des liens pour les emplacements qui peuvent fournir un meilleur éclaircissement que les miens.

2. **Algorithme MiniMax**

Le minimax (parfois minmax) est une règle de décision utilisée en théorie de la décision, théorie des jeux rectangulaires ,Pour une vaste famille de jeux à somme nulle, statistiques et philosophie pour réduire au maximum la perte possible. Alternativement, il peut considérer en tant que maximum du gain minimum (maximin). À l'origine formulé pour la théorie des jeux rectangulaires de somme nulle à deux joueurs, couvrant les deux les cas où les joueurs prennent des mouvements alternatifs et ceux où ils entreprennent des démarches simultanées. Elle a été

également prolongée à des jeux plus complexes et à la prise de décision générale en présence de l'incertitude.

Le théorème du minimax est elaboré en 1928 PAR Von Neumann **,** pour un nombre limité de coups ,). cette technique amène l'ordinateur à passer en revue toutes les possibilités pour un nombre limité de coups et à leur assigner une valeur qui prenne en compte les bénéfices pour le joueur et pour son adversaire. Le meilleur choix étant alors celui qui maximise ses bénéfices tout en minimisant ceux de son adversaire(le jeu est à somme nulle).

En limitant le nombre de nœuds visités dans l'arbre de jeu ,il existe différents algorithmes basés sur MinMax et permettant d'optimiser la recherche du meilleur coup. le plus connu dans ces algorithmes est l'élagage alpha-beta. En pratique, comme dans le cas du jeu d'échecs l'arbre est souvent trop vaste pour pouvoir être intégralement exploré. Seul une fraction de l'arbre est explorée.

```
Function minimax(node, depth)
    if node is a terminal node or depth = 0
        return the heuristic value of node
    else
        let α := -∞
        foreach child of node      { evaluation is identical for both players }
            let α := max(α, -minimax(child, depth-1))
        return α
```

a. **Principe**

pour faire remonter à la racine une valeur (appelée « valeur du jeu ») on doit visite l'arbre de jeu, cette valeur est calculée récursivement de la façon suivante :

- MinMax(p) **=**f(p) si p est une feuille de l'arbre ou f est une fonction d'évaluation de la position du jeu.
- MinMax(p) =MAX(MinMax(O_1),… ,MinMax(O_n)) si p est un nœud joueur avec fils $O_1,\ldots,O_n$.
- MinMax(p) =MAX(MinMax(O_1),… ,MinMax(O_n)) si p est un nœud opposant avec fils $O_1,\ldots,O_n$.

Exemple :

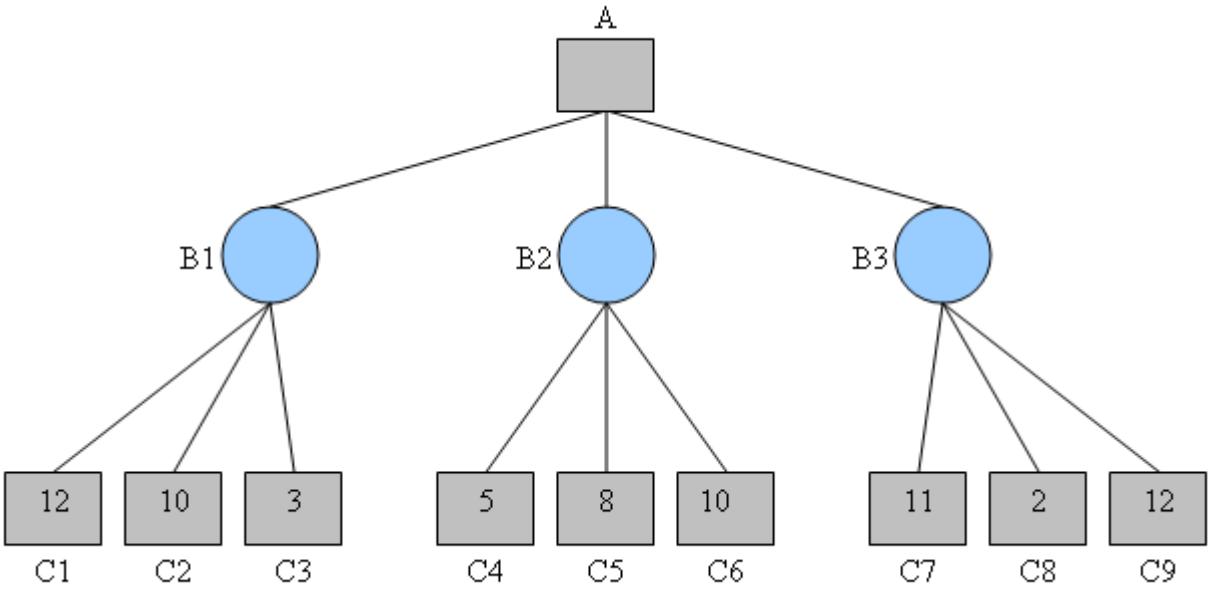


figure :Exemple .

Dans la figure ( ) ci dessous, les nœuds joueurs sont représentés par des nouds gris et les nœuds opposants sont représentés par des nouds bleus. Pour déterminer la valeur finale du nœud A, on doit choisir la valeur maximum de l'ensemble des nœuds B puisque A est un nœud joueur. Pour cela il faut donc déterminer les valeurs des nœuds opposants B qui reçoivent chacun la valeur minimum stockée dans leurs fils. La fonction d'évaluation peut alors calculer les valeurs des feuilles.

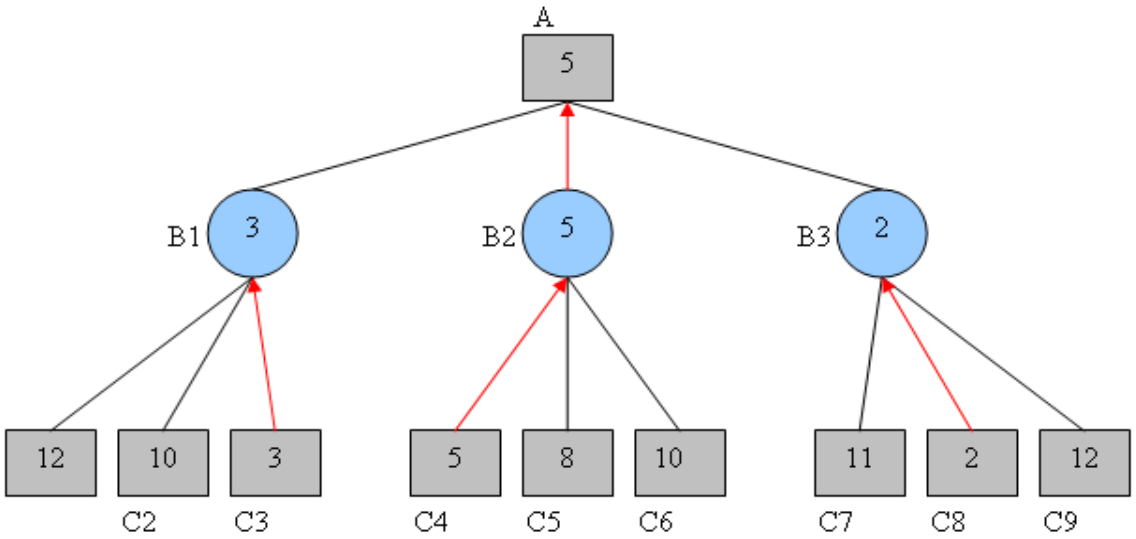


Figure : solution.

Le joueur doit jouer le coup l'amenant en B2 pour que Le nœud A prend la valeur 5. Il est claire qu'en observant l'arbre, on comprend bien que l'algorithme considère que l'opposant va jouer de manière optimale, donc il prend le minimum.

Sans ce prédicat, on choisirait le nœud C1 puisqu'il propose le plus grand gain(12) et le prochain coup sélectionné nous amène dans le nœud B1. Mais avec ce choix on prend le risque que l'opposant joue pour C3 qui propose seulement un gain de 3.

En pratique, on ne pourra pas généralement calculer la valeur théorique de la position P. Par conséquent, la fonction d'évaluation sera appliquée sur des positions non terminales. pour un meilleur résultat du calcul , il faut appliquer la fonction d'évaluation le plus loin de la racine. C'est-à-dire pour avoir un meilleur mouvement il faut examiner plus de coups successifs, ce qui permet d' obtenir une meilleure approximation de la valeur théorique

le moteur choisi.

1. Choix du programme d'échecs de freeware par Don Cross []

Bien qu'il ne soit pas dans la même ligue des programmes commerciaux, notre choix s'est derigé sur ce programme qui joue un jeu assez fort. Chenard est écrit totalement dans C++ et le code source est adaptable à tout système d'exploitation. Le code a une bibliothèque d'ouverture intégrée, une tactique spéciale de fin de jeu et un algorithme habituel de recherche minute-maximum. L'un des dispositifs le plus intéressant de Chenard est qu'il apprend automatiquement de nouvelles ouvertures et ne doit pas passer le temps à réfléchir quand est ce qu'il rencontre n'importe quelle position dans les 15 premiers mouvements d'un jeu.

On a utilisé le c++ source code de chenard qui permet de compiler et adapter l'algorithme.

Conclusion

Après ce qu'on a pu réaliser au cours de ces quatre premiers chapitres, on dispose maintenant de tout ce qui est nécessaire pour aboutir à commander un robot capable de faire des parties d'échecs.

Chapitre Cinque : Intégration.

Dans le premier chapitre, nous avons défini quelques caractéristiques du robot. Dans le second, nous avons étudié la théorie de contrôle du robot. Dans le troisième, nous avons put faire bouger le robot avec c++, et dans le quatrième, nous avons choisi un moteur de jeux d'échec qui convient le mieux à notre travail.

Le but de ce chapitre est, intégrer toutes ces données dans l'objectif de faire un jeu d'échec. Pour atteindre ce but, il faudrait tout d'abord choisir les dimensions de l'échiquier et des pièces d'échec ensuite déduire les dimensions de la pince puis, enregistrer toutes les possibilités du mouvement du robot pour faire, par la suite, une fonction qui parcourt toutes les positions suivant les entrées.

Il ne faut pas oublier, de faire deux autres fonctions pour l'ouverture et la fermeture de la pince. Enfin, on doit adapter et intégrer ces fonctions dans un programme.

I. Choix des dimensions de l'échiquier:

Pour fixer les dimensions de l'échiquier, il est nécessaire de choisir un plan bien défini sur lequel on va mettre notre plaque .Il faut au même temps voir les limites du travail de la pince dans ce même plan pour pouvoir choisir la plus grande plaque possible afin que le robot puisse bien se balader entre les pièces d'échec.

Il ne faut pas oublier que la pince doit être toujours en position horizontal pour pouvoir manipuler les pièces en toute liberté sans toucher les pièces qui sont à coté.

Après avoir effectué plusieurs essais dans des plants différents, on a trouvé que le plan de fixation du robot nous offre la plus grande surface de travail.

Dans ce plan, le robot peut tourner à droite et à gauche de 160°et il trace deux cercles de rayon maximum 855mm et le rayon minimum est 420mm comme le montre la figure ci après.

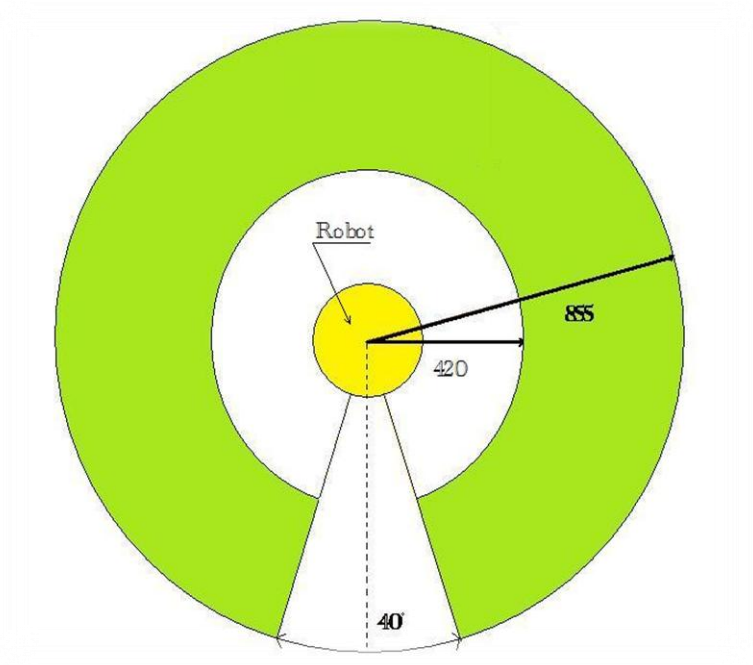


Figure : surface de travail

Pour choisir l'échiquier il faut profiter le maximum de cette surface de travail : L'échiquier ne doit pas dépasser la surface verte et il faut qu'il soit le plus grand possible .donc il faut chercher le plus grand carré à l'intérieur de cette surface.

D'après la surface obtenue on peut faire un carré de dimension 415mm×415mm.

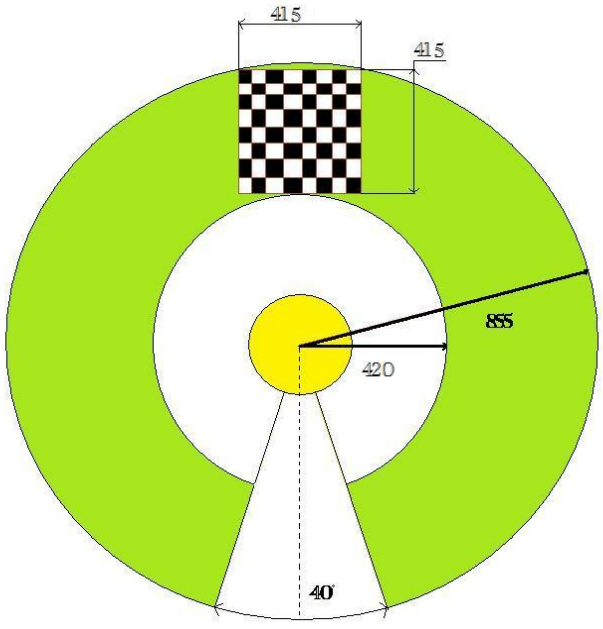


Figure : le carré maximum à l'intérieur de la surface de travail.

Pour simplifier le découpage en petits carré et avoir un travail meilleur du robot (des difficultés à atteindre les points limites) on prend un carré de dimension 400mm×400mm.

Donc on va avoir des petits carrés de 50mm×50mm pour mettre les pièces.

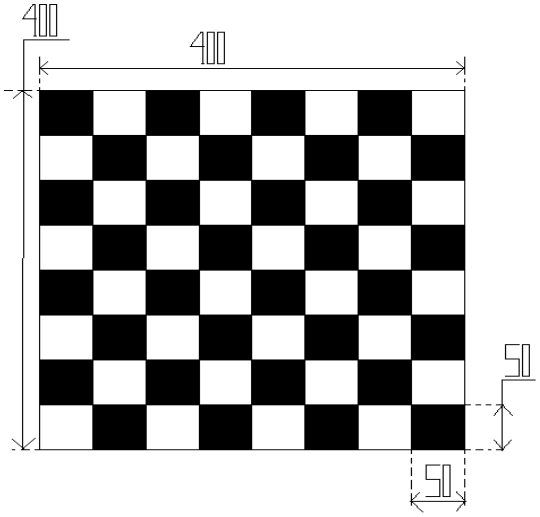


Figure : dimensions de l'échiquier.

Choix de la pince et du diamètre des pièces

Dans le paragraphe précédent on a pu dimensionner l'échiquier ainsi que les dimensions de ces carrés. Maintenant, en utilisant ce résultat, il nous reste de choisir la pince et la dimension des pièces.

Pour travailler dans ce carré, la pince du robot ne fera pas l'affaire, car elle est très grande et en plus son ouverture elle aussi est grande (220mm), ce qui nous oblige à faire des petites pièces de manière que ces derniers ne soient pas gênés par la pince. Et en plus de tout ça, il faut ajouter une adaptation à l'intérieur de la pince pour qu'elle puisse attacher les pièces.

Malgré ces modifications, ca va être très serré et la moindre faute peut tomber une pièce et gâcher le jeu.

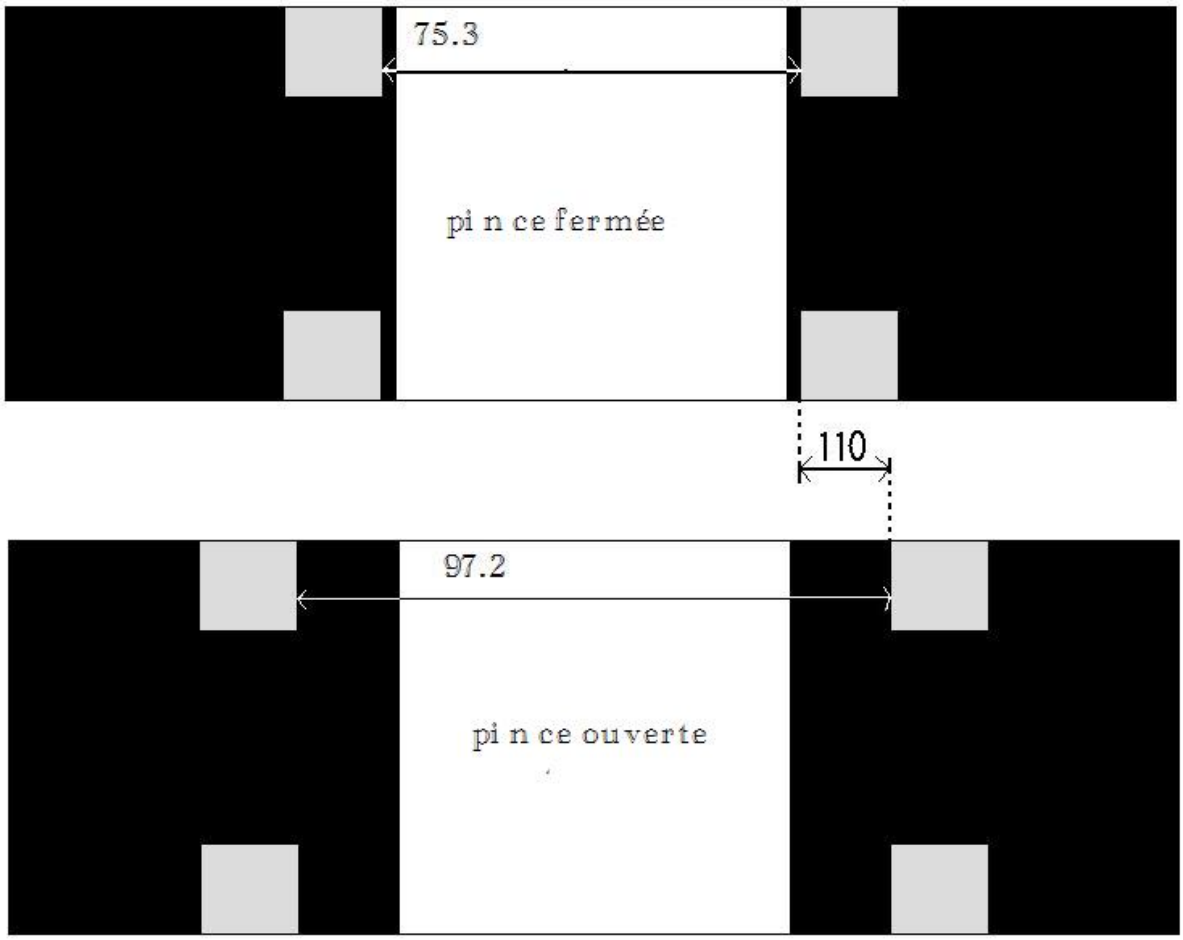


Figure : Espace occupé par la pince lors de l'ouverture et la fermeture

Donc pour un travail meilleur de la pince dans l'échiquier et afin d'éviter tout problème aussi pour avoir des pièces d'échec grandes, la meilleure solution serai, de refaire la pince. Pour cela, il faudrait d'abord, choisir un diamètre commun à toutes les pièces.

1. **Choix des pièces :**

On a pris vingt millimètres (20mm) dimension du diamètre de toutes les pièces d'échec, elles ont toutes une forme cylindrique avec une base aussi cylindrique de diamètre trente huit millimètres. On fait le dessin de toutes les pièces en solidworks,

- Quelques pièces en Solidworks

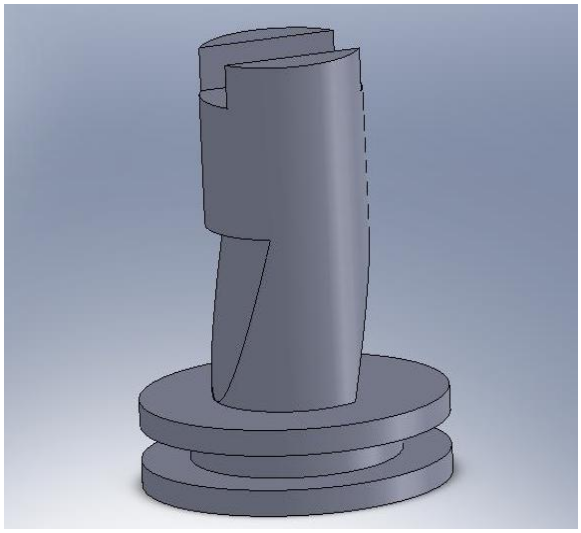


Figure : Cavalier

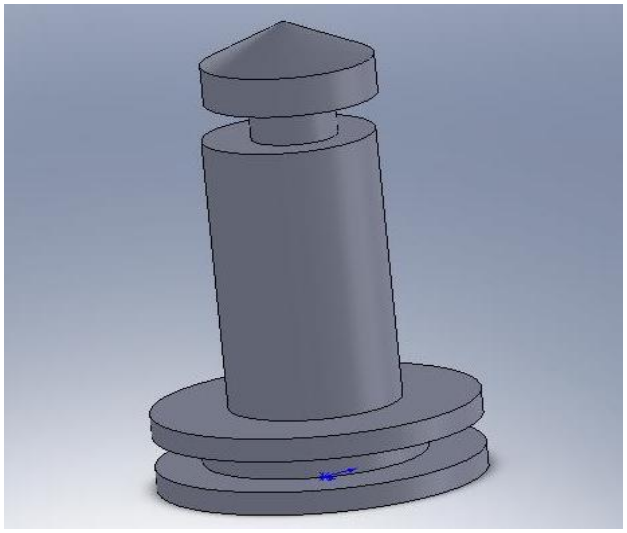


Figure : Le Roi

2. **Choix de la pince**

Après avoir choisi la forme et les dimensions des pièces, il nous reste à faire de fixer la forme et les dimensions de la pince.

Il est évident que la forme intérieur du pince là où on va attacher les pièces sera cylindrique et de diamètre égale à vingt (20mm) .Notre but est, d'avoir une pince qui peut non seulement prendre les pièces cylindrique mais aussi lorsqu'elle s'ouvre, elle ne gène pas les pièces voisines.

La pince doit être comme indique la figure (), Serre bien les pièces à la fermeture, et laisse un bon espace entre les pièces à l'ouverture.

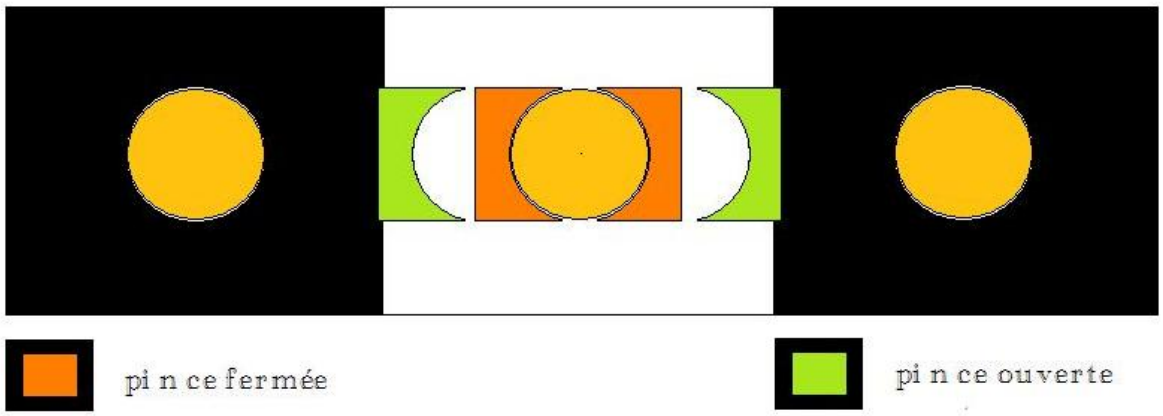


Figure : Forme de la pince avec les pièces.

Après tout calcul fait, on a choisi les dimensions sur la figure ().

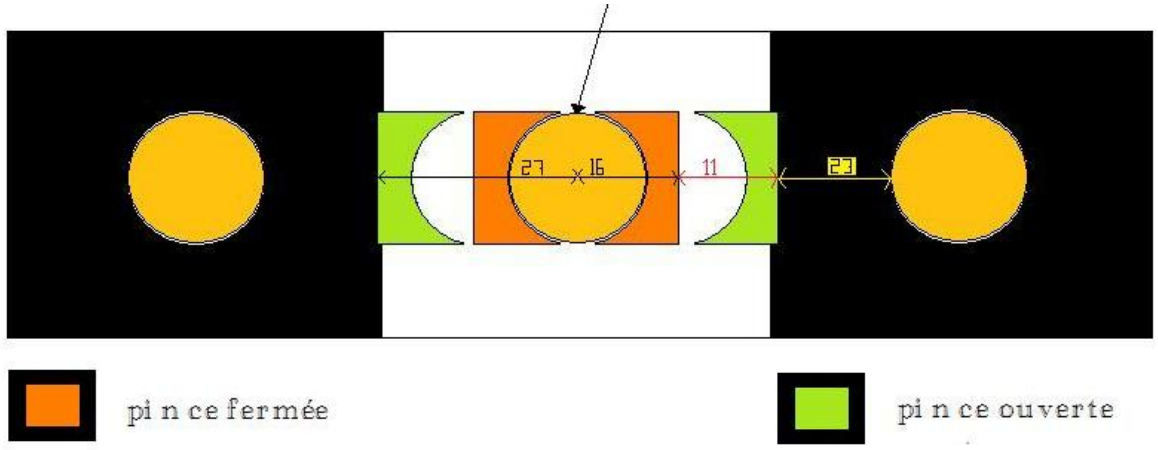


Figure : Dimensions de la pince

Pour simplifier l'usinage de la pince, on va diviser la pince en deux parties : la première partie fig () va être montée sur le mécanisme à la place de l'ancienne pince, la deuxièmes fig () va être montée sur la première partie et elle va avoir la forme des pièces.

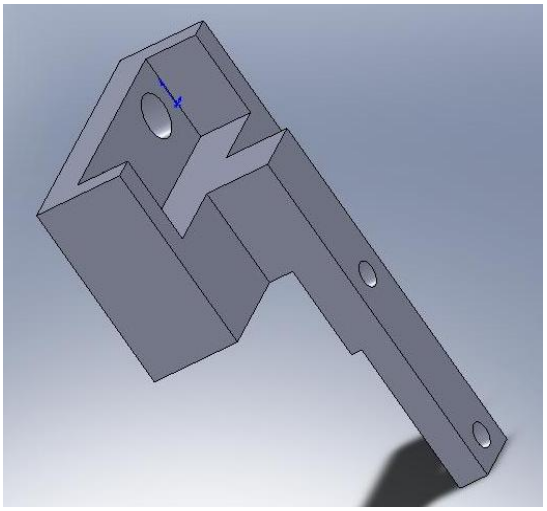


Figure : première partie de la pince

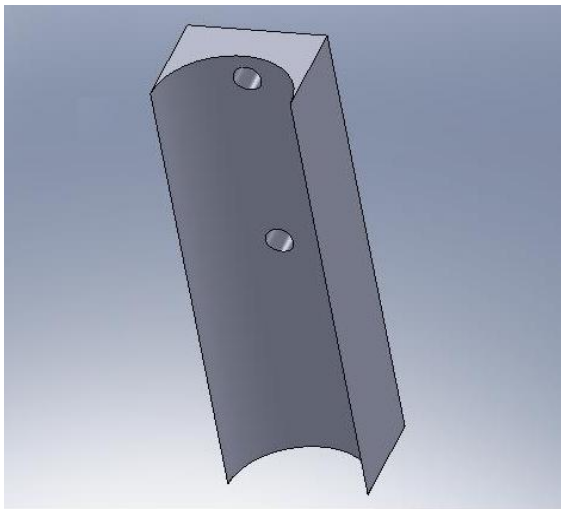


Figure : deuxième partie de la pince.

L'orsqu'on assemble les pièces on obtient notre pince qui a la forme suivante fig().

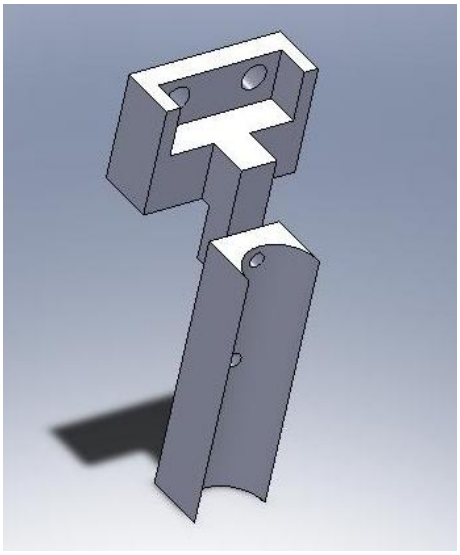


Figure : Nouvelle pince

- ❖ Pour plus de détailles sur les dimensions il faut Voir hhhhhhhhhh.

Enregistrer les locations :

1. Méthode d'enregistrement

Pour que le robot puisse jouer au échec il faut qu'il connait les soixante quatre positions dans l'échiquier .Donc le premier travail à faire est, de tenir un pion ou une autre pièce avec la pince et la faire passer par toutes les positions et les enregistrer. Il ne faut pas oublier de faire une position en haut de chaque carré pour le dégagement de la pièce dans ce carré.

Lorsqu' on somme tout les positions à enregistrer, on obtient, cent vingt huit positions à enregistrer.

- Pour les positions basses.

On a choisie de considérer l'échiquier comme un repère orthonormé dont les axes sont X et Y comme le montre la figure () .chaque position est nommée suivant ces coordonnées par rapport au repère de l'échiquier. Tout les noms commence par ' #' suivit de' L' qui signifie que c'est une position bas puis le numéro de la position.

Pour le numéro, le chiffre des dizaines est la position suivant Y et le chiffre des unités est la position suivant X.

- Pour les positions hautes il suffit de changer L par H qui signifie haut.

EXEMPLE :

#L56 : C'est une position basse, le carré numéro 56 (X=6 et Y=5).

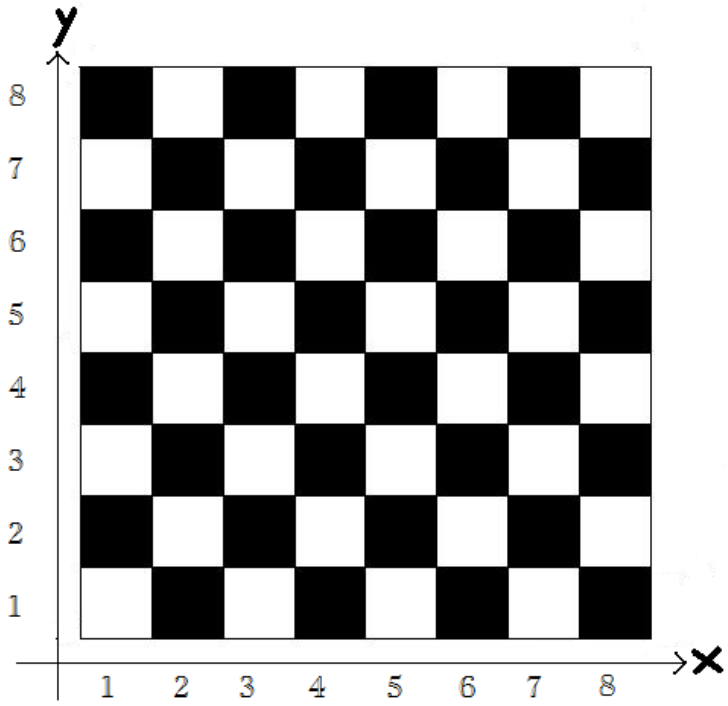


Figure : comment nommer les carrés.

2. **Déchiffrer les locations avec HHD free-serial-port-monitor**

Après avoir enregistré toutes ces locations, on ouvre le logiciel d'écoute **HHD free-serial-port-monitor** et on fait l'exécution de toutes les locations.

Chaque position va être enregistrée comme suit :

18 80 00 C0 CF 80 01 00 00 08 FE 21 0C 44 4F 20 6D 6F 76 65 20 23 6C 31 31 07 3E

En regardant toutes les locations, on remarque que seul les chiffres rouges changes et les autres restes toujours fixes.

Au lieu de faire une fonction pour chaque location, on a eu l'idée avec ce résultat de construire six fonctions de F1 à F6 qui donne le bon chiffre l'orsqu'on lui donne X et Y. F1 donne le premier, F2 le second et etc.

Puis, on a fait une fonction bougerDOWN qui a comme entrés F1,F2 à F6. Envois les dis requêtes de dialogue puis suivant les fonctions choisies, elle se déplace vers la location voulue.

Enfin, pour avoir une fonction ayant comme entrées X et Y. On a fait une fonction moveDOWN qui reçoit X et Y a l'entrée, exécute les six fonctions puis elle exécute la fonction bouger bougerDOWN.

On a répété le même travail pour les positions hautes, on a changé seulement, les noms et le plus important une fonction moveUP qui commande les pièces en haut.

Il nous reste enfin de faire une fonction ouvrir qui ordonne l'ouverture de la pince et de la même manière une fonction fermer qui ordonne la fermeture de la pince.

On a fait une fonction à zéro entrée qui exécute les dix lignes de dialogue puis elle donne l'ordre de fermer ou d'ouvrir.

3. **Remarque :**

Toutes les locations sont enregistréess avec une vitesse de 10% de la vitesse maximum.

Si on change cette vitesse et on enregistre la position ensuite on fait le codage et on envoie les dix requêtes de dialogue puis on envoie cet ordre, Le robot ne va pas bouger.

Donc, si on veut augmenter la vitesse du mouvement à une vitesse supérieur, il faut refaire les dis requêtes de dialogue avec cette vitesse choisie et refaire les locations avec la même vitesse.

Avec une vitesse égale à 10% de la vitesse max on peut avoir deux ou plusieurs affiches pour une même location dans le logiciel d'écoute, ces requêtes mènent tous à faire bouger le robot.

Exemple :

On a écouté l'envoie du commande **MOVE #zero** deux fois de suite et on a obtenue une différence en trois chiffres qui sont en rouge.

Le premier enregistrement :

19 80 00 C0 CF 80 01 00 00 00 5C 21 0D 44 4F 20 4D 4F 56 45 20 23 7A 65 72 6F 07 07

Le second:

19 80 00 C0 CF 80 01 00 00 00 4B 21 0D 44 4F 20 4D 4F 56 45 20 23 7A 65 72 6F 06 F6

4. Conclusion :

Il faut respecter la vitesse lorsqu'on veut bouger le robot.

Pour une seule location le logiciel SRCwin peut envoyer à chaque fois une requête différente avec une même vitesse.

Intégration de toutes les fonctions avec le moteur d'échec :

Dans le chapitre quatre on a put choisir un moteur d'échec, et dans ce chapitre, notre but est d'adapter ce logiciel au mouvement du robot. Chaque mouvement des pièces dans l'écran doit être accompagné par un mouvement effectué par le robot des pièces sur l'échiquier. Donc il faut chercher ou intégrer les fonctions de mouvement dans ce moteur.

Pour que l'humain bouge ses pièces, il doit indiquer la position initiale et finale de la pièce à déplacer. Avec ces informations, on peut adapter nos fonctions pour bouger les pièces. Mais le problème est au niveau de l'ordinateur qui nous donne que ca position finale. Et c'est ici ou réside le problème.

Lorsqu'on revient à l'affichage dans l'ordinateur, on voit que le moteur affiche FEN après tout mouvement, de l'humain ou de l'ordinateur .il faut essayer de trouver une solution avec cet affichage.

1. Observer ce que le logiciel fait:

- Affichage initial de FEN :

```
FEN = rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR
```

```
+--------+
¦rnbqkbnr¦8
¦pppppppp¦7
¦........¦6
¦........¦5
¦........¦4
¦........¦3
¦PPPPPPPP¦2
¦RNBQKBNR¦1
+--------+
 abcdefgh
```

- Demande de bouger une pièce blanche :

```
White move>
```

- Bouger un pion de g2 à g4 :

```
White move> g2g4
```

- Affichage de FEN devient :

```
FEN = rnbqkbnr/pppppppp/8/8/6P1/8/PPPPPP1P/RNBQKBNR
```

```
+--------+
¦rnbqkbnr¦8
¦pppppppp¦7
¦........¦6
¦........¦5
¦......P.¦4
¦........¦3
¦PPPPPP.P¦2
¦RNBQKBNR¦1
+--------+
 abcdefgh
```

- Le mouvement de l'ordinateur et l'affichage de FEN :

```
Black: d5
```

```
FEN = rnbqkbnr/ppp1pppp/8/3p4/6P1/8/PPPPPP1P/RNBQKBNR
```

```
+--------+
¦rnbqkbnr¦8
¦ppp.pppp¦7
¦........¦6
¦...p....¦5
¦......P.¦4
¦........¦3
¦PPPPPP.P¦2
¦RNBQKBNR¦1
+--------+
 abcdefgh
```

2. **Remarque :**

On constate dans cet exemple que FEN affiche l'emplacement des pièces à chaque mouvement. L'affichage commence à décrire l'état de la ligne huit jusqu'a arriver à la ligne une et entre chaque ligne il ya ' / '.

Pour décrire le contenue d'une ligne, FEN commence de gauche à droite. S'il y a une pièce, on met sa lettre significative. Et lorsqu'on trouve du vide, on affiche le nombre des vides successifs.

3. **Solution.**

Pour connaitre la position initiale de l'ordinateur il suffit de comparer **FEN** et **FEN** de l'humain (**FEN** précédant).

On a construit un tableau **' occupancy [ ]'** qui parcourt FEN, s'il trouve une lettre il écrit cette lettre, et s'il trouve un nombre il affiche des zéro égales à ce nombre.

Maintenant pour comparer il nous faut un autre tableau qui enregistre l'état précédant de l'échiquier, ce tableau porte le nom de' Occupancy_old[ ] '. il sera initialisé avant le début de toute partie avec l'état initiale des pièces dans l'échiquier. puis il sera affectté par' occupancy[ ]' a chaque fois ou on fait bouger une pièce.

Donc le travail à faire est de comparer ' Occupancy_old[ ] 'et' occupancy[ ]'.

4. **Comparer' Occupancy_old[ ] 'et' occupancy[ ]'.**

En ce qui concerne le mouvement du robot par l'humain, on peut changer les entrées des fonctions de X et Y en une chaine de caractère qui analyse ce que l'humain donne et fait bouger le robot. En faisant cette idée on a trouvés qui si on donne un mauvais mouvement de la pièce l'ordinateur vous affiche que cette pièce ne peut pas aller à cet emplacement, par contre le robot peut bouger.

Pou cela on a décidé de bouger l'humain et l'ordinateur en faisant des raisonnements logique avec **FEN.**

Il faut savoir quoi faire en faisant une comparaison entre l'état précédant des pièces dans l'échiquier et le mouvement actuel .il faut différentier entre bouger une pièce, attaquer une pièce et rien faire lorsqu'on commence le jeu.

- Pour rien faire :

Il faut que (Occupancy_old[i] = occupancy[i])

- Pour déplacer une pièce :

Pour déplacer il faut connaitre la position initiale et finale et il faut que ces deux conditions soient présentes au même temps.

- si (Occupancy_old[i] =lettre) et (occupancy[i]=0)

- si (Occupancy_old[j] = 0) et (occupancy[j]=une lettre)

Alors i est la position initiale et j est la position finale.

- Pour attaquer :

Il faut que ces deux conditions soient présentes au même temps pour trouver la pièce qui attaque et la pièce qui sera jetée.

- Si (Occupancy_old[i] = lettre) et (occupancy[i]=0)

- Si (Occupancy_old[j] =lettre) et (occupancy[j]=lettre)

et (Occupancy_old[j] ≠ occupancy[j])

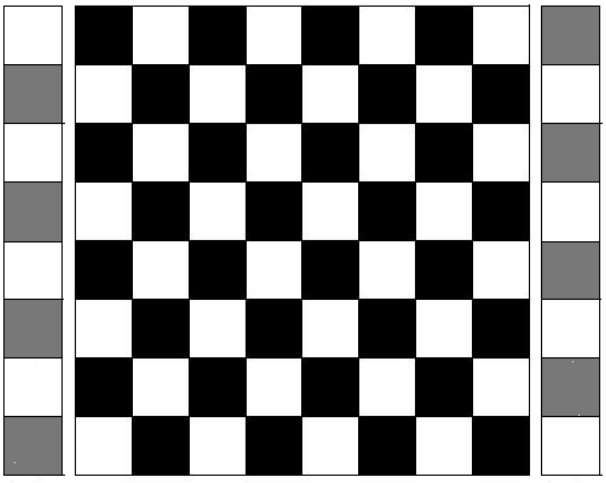
Alors i est la position initiale de la pièce qui va attaquer et j est la position de la pièce a jeté.

5. **Le pion à la ligne finale.**

Lorsqu'un pion atteint la ligne finale de l'échiquier on peut le remplacer par une pièce sortie du jeu au choix. Le robot doit donc connaitre ou mettre des pièces à jeter puisqu'on a encore besoin d'eux. Donc il doit mettre les pions dans n'importe quelle location puisque on a plus besoin d'eux. Mais pour les autres pièces, il doit mettre chaque pièce dans un endroit qu'il connait.

Pour ne pas dépasser l'espace du travail du robot on a choisi de placer les pièces blanches à gauches de l'échiquier dans la nouvelle colonne en gris et blanc dans la fig() et les noirs dans le coté opposé .pour atteindre ses carré, on a enregistré encore trente deux positions et on a fait les fonctions suivantes :

- moveUPwhite (x, y) : pour bouger les pièces blanc en haut.
- moveDOWNwhite (x, y) : pour bouger les pièces blanc en bas.
- moveUPblack (x, y) : pour bouger les pièces noirs en haut.
- moveDOWNblack (x, y) : pour bouger les pièces noirs bas.



Fig() les positions des pièces rejeter.

Il faut donner un emplacement à chaque pièce pour pouvoir la reprendre après.

Les pièces vont prendre une même disposition verticale la même que la disparition initiale. Mais le problème qui se pose est le suivant :

Comment différentier entre les pièces pareilles d'un même couleur ?

Pour résoudre ce problème on a deux tableaux le premier pour les pièces blanches et le second pour les pièces noires qui indiquent la disponibilité des carrés.

Ces tableaux affectés initialement par zéro et à chaque fois ou une pièce est mit, ce carré prend le nom du chiffre significatif de cette pièce.

Initialisation des tableaux :

Pieces_Removed_White[i]='0';

Pieces_Removed_Black[i]='0';

Pour les pièces doubles qui on deux location disponibles comme la tour, le cavalier et le fou ; Il ya vérification de la disponibilité de la première place :

si elle est vide on met la première pièce dans cette location. Sinon on est obligé de la mettre dans ca deuxième location.

Pseudo code des mouvements :

Avant d'expliquer comment le robot peut bouger, il est nécessaire de noter qu'avant tout mouvement le robot doit commencer toujours dans les positions hautes et la pince doit être ouverte. À la fin de chaque mouvement d'une pièce doit revenir à cet état (en positions hautes et la pince ouverte).

1. Pour prendre une pièce d'une position :

```
moveUP(x,y);
moveDOWN(x,y);
fermer();
moveUP(x,y);
```

2. Pour poser une pièce sur une position:

moveUP(x,y);

moveDOWN(x,y);

ouvrir();

moveUP(x,y);

3. Pour attaquer une pièce

a. Prendre la pièce qui a été attaquée

moveUP(x,y);

moveDOWN(x,y);

fermer();

moveUP(x,y);

Analyser la pièce ;

Trier les pièces

```
if (pièce = pion)
{
     positionJETTER();
     ouvrir();
}
else
{
     if(pièce = blanche)
     {
           moveUPwhite ( x, y);
           moveDOWNwhite ( x, y);
           ouvrir();
           moveUPwhite ( x, y);
     }
     else
     {
          moveUPblack ( x, y);
          moveDOWNblack ( x, y);
          ouvrir();
          moveUPblack ( x, y);
     }
}
```

Mètre la pièce qui attaque à ca place :

moveUP(x,y);

moveDOWN(x,y);

ouvrir();

moveUP(x,y);

Résultats de tout le travail.

Dans ce paragraphe on va s'intéresser à justifier le bon fonctionnement de notre projet.

Pour cela nous allons en premier lieu prouver le bon choix des instruments du jeu (échiquier et pince). Ensuite, nous allons montrer que le robot peut bouger les pièces avec c++. Enfin, on montrera que le robot suit et en coordinance avec le moteur de jeu d'échecs.

1. **Le bon choix des de l'échiquier, la pince et les pièces.**

a. **Le robot peut atteindre les positions limites de l'échiquier.**

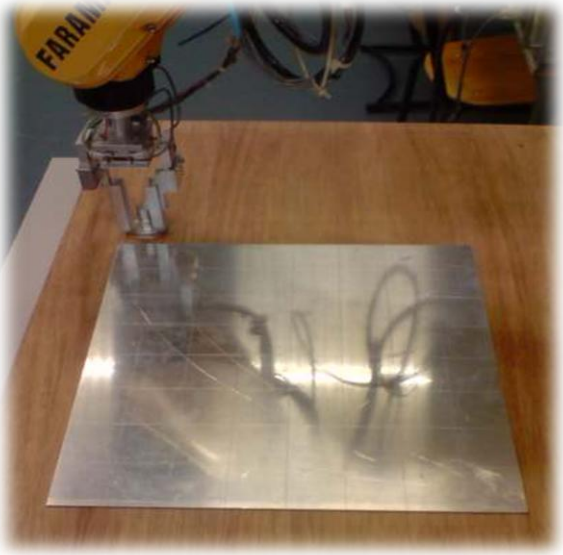


Figure :la position à l'exterme gauche.

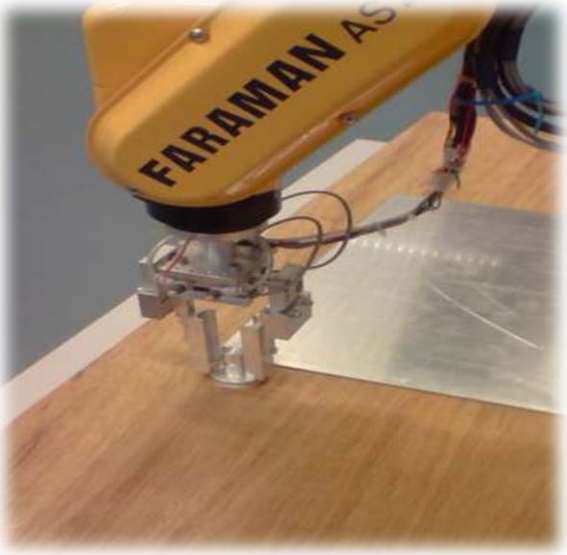


Figure :la position à l'extreme droite.

Montrer qu'il tient bien les pièces.

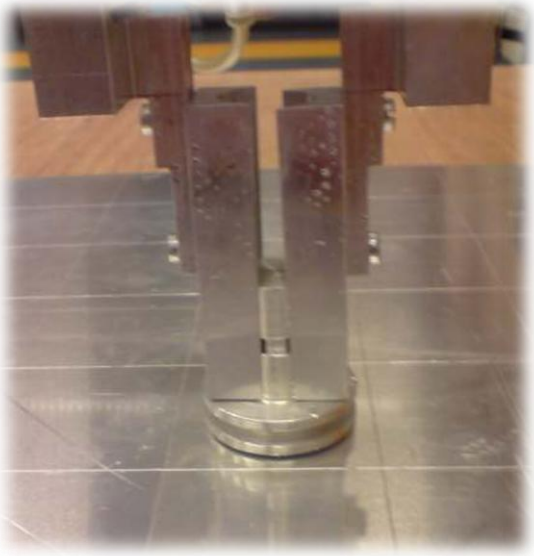


Figure :pince tient bien la piece

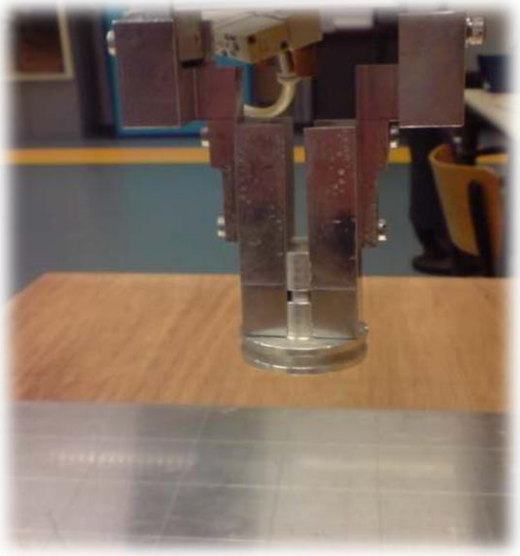


Figure :la pince prend la piece sans qu'elle tombe

Montrer que la pince ne touche pas les autres pièces à l'ouverture et à la fermeture.

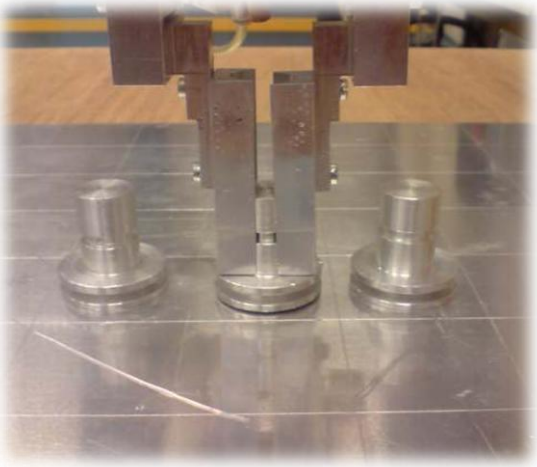


Figure :pince fermeé

Figure :pince ouverte

2. **Le robot commandé avec c++ .**

le robot peut bouger a tout les positions enregistrées et ces positions on peut les visualiser avec le logiciel d'écoute sur port série.

a. **Bouger le robot avec c ++.**

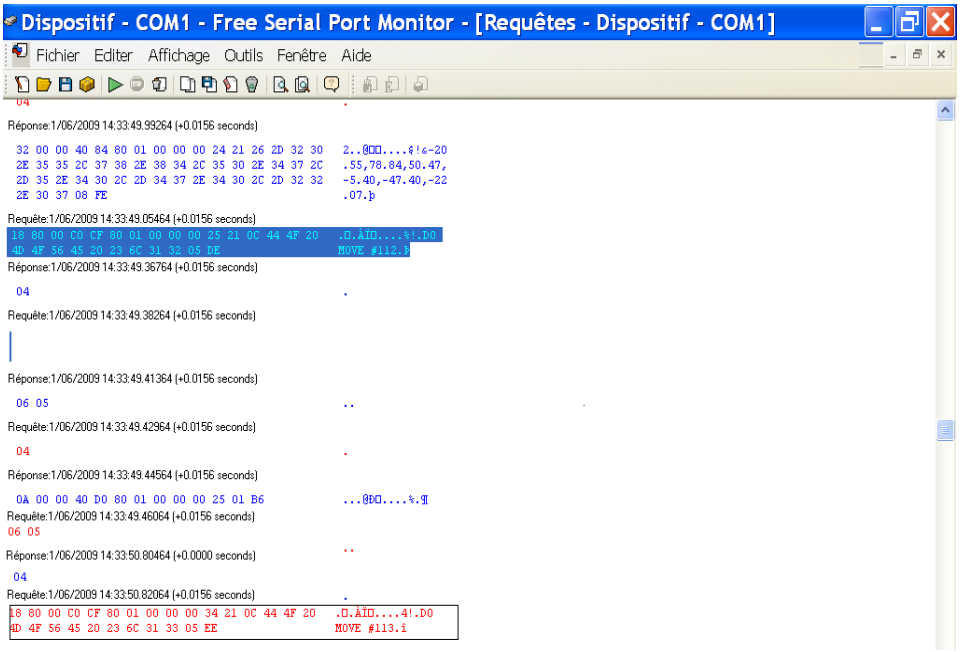


Figure :le logiciel d'ecoute d'etecte le mouvement vert les position 13 et 12.

Ouvrir la pince avec c++.

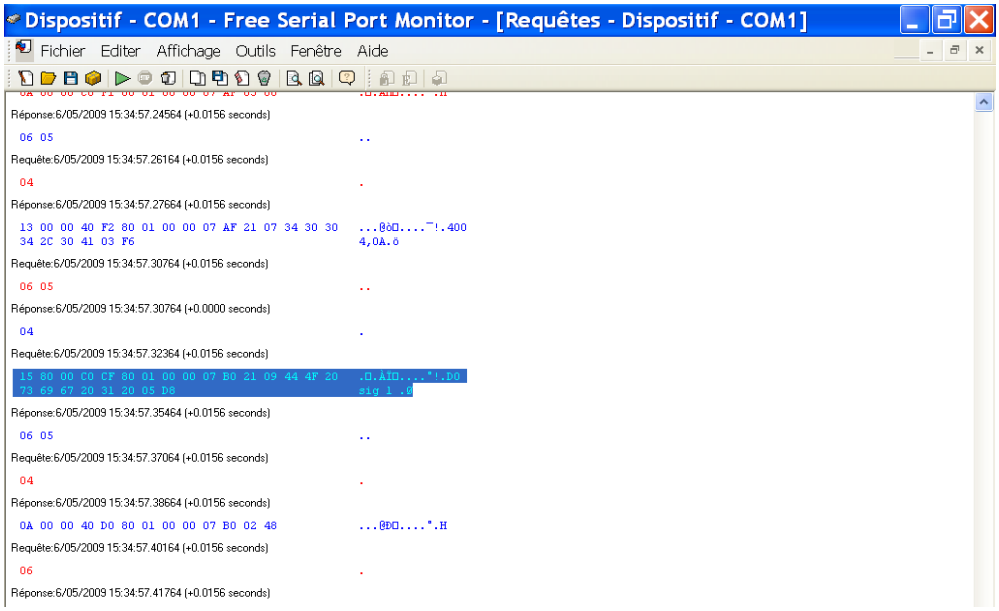


Figure :le logiciel d'ecoute d'etecte l'ouverture de la pince.

fermer la pince avec c++.

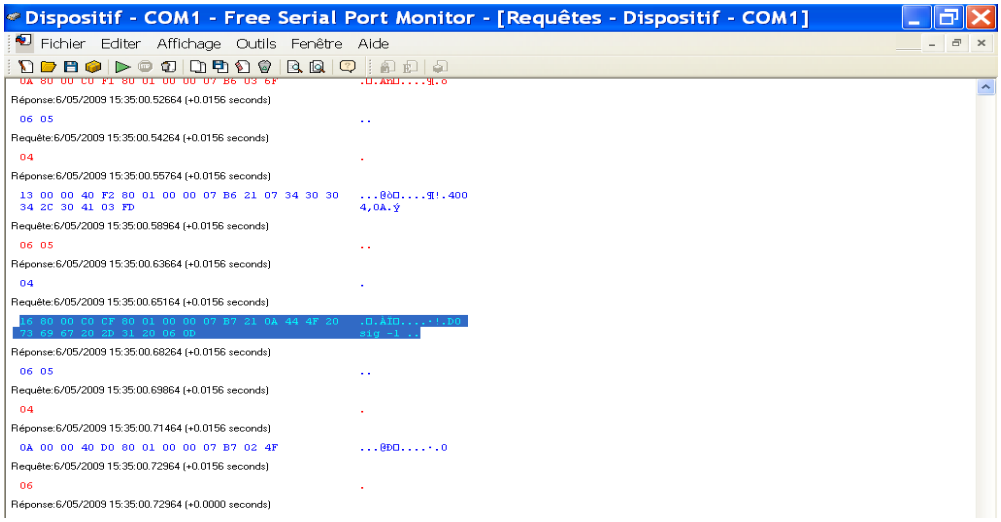


Figure :le logiciel d'ecoute d'etecte la fermiture de la pince.

3. **Le robot et le moteur de jeu d'échec.**

Pour jouer a l'échec le robot doit déplacer, attaquer jeter les pions perdus et garder les autres pièces.

En attaquant une pièce on peut voire tout ce qui a précédés.

a. **Prendre la pièce à jeter.**

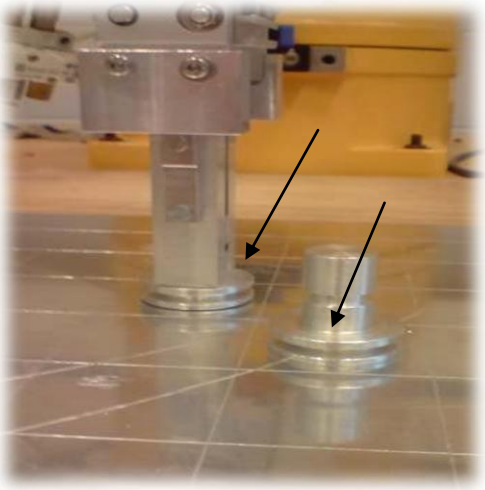


Figure :piece 2 attaque piece 1 .

Eloigner la pièce perdue.

- Si c'est un pion on le jet.

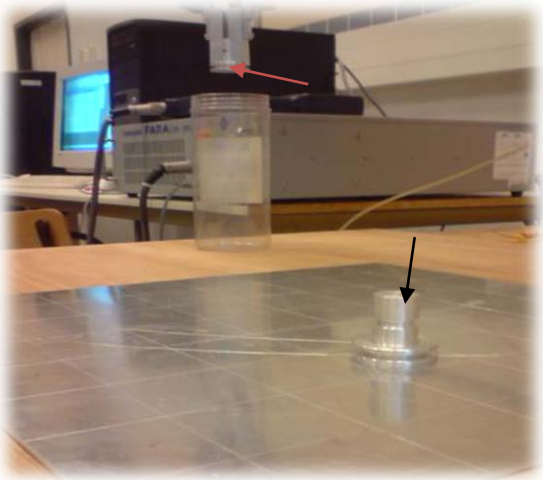


Figure : pion 1 jeter.

- Si non on garde cette pièce.

Figure : piéce 1 gardé.

On prend la pièce qui attaque

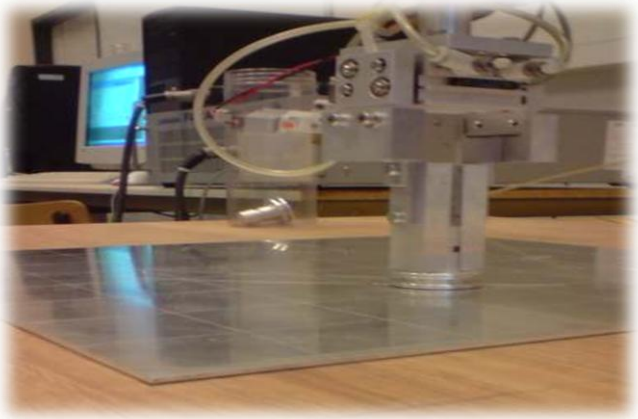


Figure :Prendre la piece 2.

On la met à la place de la pièce perdue.

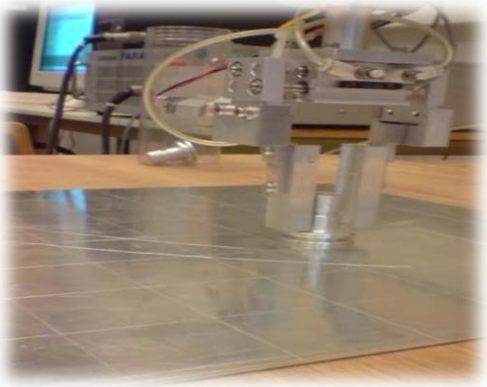


Figure : Mettre la piece 2 a la place de la piece 1 .

4. **Conclusion :**

Le robot execute tout les commandes qui ont été faites par le programme de jeu d'echec.

On peut maintenat jouer a l'echec contre le robot.[3]

Conclusion et perspectives

Ce livre décrit bien le robot SAMSUNG FARAMAN AS2 ainsi que les étapes suivies pour aboutir a notre objectif. Qui est l'adaptation des commandes données a un moteur de jeu d'échec pour que le robot puisse manœuvrer les pièces de l'échiquier.

Il faut noter que les objectifs de ce projet ont été atteints puisque notre robot n'a pas trouvé aucune difficulté de coordination avec le moteur du jeu d'échec.

Il faut aussi signaler que la méthode choisie pour balayer les locations n'est pas très efficace car si on change l'échiquier tout les locations enregistres seront perdues alors il vaut mieux que se soit le robot qui reconnaît les positions voulues sans pour autant qu'il les'apprenne. Et pour cela, plusieurs solutions électroniques peuvent être montées au robot et à l'échiquier pour éviter ce problème.

Ce travail m'a été d'un grand apport Pour mon expérience professionnel car il ma permis d'améliorer mes connaissances a la fois dans le langage de programmation et dans le domaine robotique, il m'a aussi donné l'opportunité de toucher de près au difficultés lors de la résolution d'un problème qu'on qualifie de complexe et d'exploiter mes connaissances théorique dans le domaine pratique.

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Bibliographie

[1]. a
[2]. k
[3]. b
[4].

fr.wikipedia.org/wiki/Fichier:Min_Max_Sample_1.png

cosinekitty.com/chenard/#source

en.wikipedia.org/wiki/Minimax

fiascochess.wordpress.com/2009/03/08/the-layout-of-a-chess-engine/

lwh.free.fr/pages/algo/minmax/minmax.htm

http://eavr.u-strasbg.fr/wiki/upload/a/a4/Cours_rob_intro.pdf

www.robot-maker.com/robotscope/robot-mobile.jpg

Manuel de l'utilisateur

www.vieartificielle.com/images_nouvelle/vic6000.1.jpg

/www.gizmodo.fr/wp-content/uploads/2009/03/sky_warrior.jpg

www.science-et-vie.net/img/illustrations/R/robot-asimo.jpg

www.lattis.univ-toulouse.fr/lesia/uploaded/images/7axes.jpg

www.dumatek.fr/topic2/bv000001.jpg

http://img.directindustry.com/images_di/photo-g/scara-robot-38412.jpg

www.rosier.fr/uploads/robot-cartesien-IAI.jpg

www2.cnrs.fr/sites/thema/image/3ecdc6caab06d.jpg

COUR ROBOTQUE PROF BAUDOIN

www.virtualserialport.com/products/serialmonitor/?gr=fpm01&gclid=CKyT04TJ5poCFUEA4woddXOQBA

www.fogma.co.uk/foggylog/archive/140.html

Q=queen,

R= rook,

B= bishop

N= knight

P= pawn

K=KING